

DBMS – Assignment-B3 – Softcopy

Queries

1. Return the Total Salary of per Company

```
db.Employee.mapReduce(  
    function() {  
        emit(this.Company_name, this.Salary);  
    },  
    function(key, values) {  
        return Array.sum(values);  
    },  
    { out: "total_salary_per_company" }  
);  
{ result: 'total_salary_per_company', ok: 1 }  
[  
    { _id: 'Roxonn', value: 50000 },  
    { _id: 'SNASA', value: 80000 },  
    { _id: 'Vought', value: 39000 },  
    { _id: 'Lex Corp.', value: 45000 },  
    { _id: 'Wayne Industries', value: 70000 },  
    { _id: 'Stark Industries', value: 32000 },  
    { _id: 'Oscorp', value: 52000 },  
    { _id: 'Y-Space', value: 75000 }  
]
```

2. Return the Total Salary of Company Name:"Oscorp"

```
db.Employee.mapReduce(  
    function() {  
        if (this.Company_name === "Oscorp") {  
            emit(this.Company_name, this.Salary);  
        }  
    },  
    function(key, values) {  
        return Array.sum(values);  
    },  
    { out: "total_salary_oscorp" }  
);  
{ result: 'total_salary_oscorp', ok: 1 }  
[ { _id: 'Oscorp', value: 52000 } ]
```

3. Return the Avg Salary of Company whose address is “Pune”.

```
db.Employee.mapReduce(  
    function() {
```

```

        if (this.Address.some(addr => addr.PAddr.includes("Pune"))) {
            emit("Pune", this.Salary);
        }
    },
    function(key, values) {
        return Array.avg(values);
    },
    { out: "avg_salary_pune" }
);

{ result: 'avg_salary_pune', ok: 1 }
[ { _id: 'Pune', value: 48500 } ]

```

4. Return the Total Salary for each Designation of Wayne Industries.

```

db.Employee.mapReduce(
    function() {
        if (this.Company_name === "Wayne Industries") {
            emit(this.Designation, this.Salary);
        }
    },
    function(key, values) {
        return Array.sum(values);
    },
    { out: "total_salary_wayne" }
);

{ result: 'total_salary_wayne', ok: 1 }
[ { _id: 'Project Manager', value: 70000 } ]

```

5. Return total count for “State=Maharashtra”

```

db.Employee.mapReduce(
    function() {
        this.Address.forEach(function(addr) {
            if (addr.LAddr === "Maharashtra") {
                emit("Maharashtra", 1);
            }
        });
    },
    function(key, values) {
        return Array.sum(values);
    },
    { out: "count_state_maharashtra" }
);

{ result: 'count_state_maharashtra', ok: 1 }
[ { _id: 'Maharashtra', value: 2 } ]

```

6. Return Count for State Telangana and Age greater than 40.

```
db.Employee.mapReduce(  
    function() {  
        if (this.Age > 40) {  
            this.Address.forEach(function(addr) {  
                if (addr.LAddr === "Telangana") {  
                    emit("Telangana_Age_Above_40", 1);  
                }  
            });  
        }  
    },  
    function(key, values) {  
        return Array.sum(values);  
    },  
    { out: "count_telangana_age_above_40" }  
);  
{ result: 'count_telangana_age_above_40', ok: 1 }  
NONE
```

--- END OF DOCUMENT ---