B3

**1. Return the Total Salary of Each Company**

* **Map Function**: The map function emits each company name and the corresponding salary.

var mapFunction = function() {

emit(this.CompanyName, this.Salary);

};

* **Reduce Function**: The reduce function sums up the salary for each company.

var reduceFunction = function(company, salaries) {

return Array.sum(salaries);

};

db.Employee.mapReduce(mapFunction, reduceFunction, { out: "totalSalaryPerCompany" })

**2. Return the Total Salary of Company Name "TCS"**

* **Map Function**: Emit only if the company is "TCS".

var mapFunctionTCS = function() {

if (this.CompanyName === "TCS") {

emit(this.CompanyName, this.Salary);

}

};

* **Reduce Function**: Sum the salaries for "TCS".

var reduceFunctionTCS = function(company, salaries) {

return Array.sum(salaries);

};

db.Employee.mapReduce(mapFunctionTCS, reduceFunctionTCS, { out: "totalSalaryTCS" })

**3. Return the Average Salary of a Company Whose Address is "Pune"**

* **Map Function**: Emit only if the company address is "Pune".

var mapFunctionPune = function() {

if (this.Address.PAddr === "Pune") {

emit(this.CompanyName, this.Salary);

}

};

* **Reduce Function**: Calculate the average salary.

javascript

Copy code

var reduceFunctionPune = function(company, salaries) {

return Array.avg(salaries);

};

db.Employee.mapReduce(mapFunctionPune, reduceFunctionPune, { out: "avgSalaryPune" })

**4. Return the Total Salary for Each Designation of Employees in Infosys**

* **Map Function**: Emit the designation and salary for employees of Infosys.

var mapFunctionInfosys = function() {

if (this.CompanyName === "Infosys") {

emit(this.Designation, this.Salary);

}

};

* **Reduce Function**: Sum the salaries for each designation in Infosys.

var reduceFunctionInfosys = function(designation, salaries) {

return Array.sum(salaries);

};

db.Employee.mapReduce(mapFunctionInfosys, reduceFunctionInfosys, { out: "totalSalaryInfosys" })

**5. Return Total Count for "State=AP"**

* **Map Function**: Emit a count for employees where the state is "AP".

var mapFunctionAP = function() {

if (this.Address.State === "AP") {

emit("State\_AP", 1);

}

};

* **Reduce Function**: Count the total number of employees in "AP".

var reduceFunctionAP = function(state, counts) {

return Array.sum(counts);

};

db.Employee.mapReduce(mapFunctionAP, reduceFunctionAP, { out: "countAP" })

**6. Return Count for State "AP" and Age Greater Than 40**

* **Map Function**: Emit a count for employees where the state is "AP" and age is greater than 40.

var mapFunctionAPAge = function() {

if (this.Address.State === "AP" && this.Age > 40) {

emit("State\_AP\_Age\_gt\_40", 1);

}

};

* **Reduce Function**: Count the total number of employees in "AP" with age greater than 40.

var reduceFunctionAPAge = function(state, counts) {

return Array.sum(counts);

};

db.Employee.mapReduce(mapFunctionAPAge, reduceFunctionAPAge, { out: "countAPAgeGt40" })