Practical no. 4

Implement Map-reduce operation with suitable example using MongoDB.

```
1) db.Employee.insertMany(
... [
...     { emp_id: 101, name: "Ravi Kumar", department: "Sales", salary: 65000 },
...     { emp_id: 102, name: "Priya Sharma", department: "IT", salary: 82000 },
...     { emp_id: 103, name: "Amit Joshi", department: "Sales", salary: 60000 },
...     { emp_id: 104, name: "Neha Singh", department: "HR", salary: 55000 },
...     { emp_id: 105, name: "Vikram Mehta", department: "IT", salary: 95000 }
... ]
... );
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('69158f294a39d95fa763b112'),
    '1': ObjectId('69158f294a39d95fa763b113'),
    '2': ObjectId('69158f294a39d95fa763b114'),
    '3': ObjectId('69158f294a39d95fa763b115'),
    '4': ObjectId('69158f294a39d95fa763b116')
  }
}

2) Map Function
var mapFunction = function()
... {
... emit(this.department, { salary: this.salary, count: 1 });
... };

3) Reduce Function
var reduceFunction = function(key, values)
... {
... var reducedObject = { salary: 0, count: 0 };
... for (var i = 0; i < values.length; i++)
... {
... reducedObject.salary += values[i].salary;
... reducedObject.count += values[i].count;
... }
... return reducedObject;
... };

3) Map_Reduce Function execution:
db.Employee.mapReduce(
... mapFunction,
... reduceFuunction,
... {
... out: "total_salary_report"
... }
... );
{ result: 'total_salary_report', ok: 1 }

4) db.total_salary_report.find()
[
  { _id: 'HR', value: { salary: 55000, count: 1 } },
  { _id: 'Sales', value: { salary: 125000, count: 2 } },
  { _id: 'IT', value: { salary: 177000, count: 2 } }
]
```