|  |
| --- |
| **Problem Statement:** Implement Map reduces operation with suitable example using MongoDB. |

**Source Code:**

|  |
| --- |
| db.sales.insertMany([  { "\_id": 1, "product": "Laptop", "category": "Electronics", "amount": 500, "date": new Date("2024-03-01") },  { "\_id": 2, "product": "Shirt", "category": "Clothing", "amount": 200, "date": new Date("2024-03-02") },  { "\_id": 3, "product": "Smartphone", "category": "Electronics", "amount": 700, "date": new Date("2024-03-03") },  { "\_id": 4, "product": "Shoes", "category": "Clothing", "amount": 150, "date": new Date("2024-03-04") },  { "\_id": 5, "product": "Rice", "category": "Grocery", "amount": 100, "date": new Date("2024-03-05") }  ])  // Map function - emits key-value pairs  var mapFunction = function() {  emit(this.category, this.amount);  };  // Reduce function - aggregates sales per category  var reduceFunction = function(key, values) {  return Array.sum(values);  };  // Execute MapReduce  db.sales.mapReduce(  mapFunction,  reduceFunction,  {  out: "total\_sales\_per\_category"  }  );  // Display results  db.total\_sales\_per\_category.find().pretty(); |

**OutPut:**

|  |
| --- |
| > db.total\_sales\_per\_category.find().pretty();  { "\_id" : "Electronics", "value" : 1200 }  { "\_id" : "Clothing", "value" : 350 }  { "\_id" : "Grocery", "value" : 100 } |