**DBMS Practical No: 11**

**# Implement Map reduces operation with suitable example usingMongoDB.**

**Step 1: Connect to MongoDB**

Ensure you have MongoDB installed and running. Open your terminal or command prompt and run:

**bash**

**mongo**

This will open the MongoDB shell.

**Step 2: Create a Sample Collection**

Create a sample collection called sales and insert some data. For this example, we'll insert a few documents with fields like product, quantity, and revenue. Run the following commands in the MongoDB shell:

**javascript**

**use mydb db.sales.insertMany([ { product: "A", quantity: 10, revenue: 500 }, { product: "B", quantity: 5, revenue: 300 }, { product: "A", quantity: 7, revenue: 350 }, { product: "C", quantity: 3, revenue: 150 }, ]);**

Replace mydb with your database name.

**Step 3: Define the Map Function**

In the MongoDB shell, define the map function as follows:

**javascript**

**var mapFunction = function() { emit(this.product, this.revenue); };**

**Step 4: Define the Reduce Function**

Define the reduce function in the MongoDB shell:

**javascript**

**var reduceFunction = function(key, values) { return Array.sum(values); };**

**Step 5: Run the MapReduce Operation**

Execute the MapReduce operation by running this command:

**javascript**

**db.sales.mapReduce( mapFunction, reduceFunction, { out: "product\_revenue" } );**

This command will execute the MapReduce operation, and the results will be stored in a new collection called product\_revenue.

**Step 6: Retrieve the Results**

You can now query the results from the product\_revenue collection:

**javascript**

**db.product\_revenue.find();**

This will provide you with a list of products and their total revenues.