Progressive Education Society’s

**MODERN COLLEGE OF ENGINEERING, Pune -05.**

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

MCA Department

**PRACTICAL SUBMISSION RECORD- A.Y. 2024-25**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class: FYMCA Div: A**  **Semester: II** | **Course Code: MCA01554**  **Course Name: Laboratory Practice - II** | | **Batch: F2** |
| **Name: Pranav Raju Malwatkar** | | **Roll No: 51037** | |
| **CO No:** **CO517C.3** | | **Assignment No: 4** | |

**Program Title: 4)**

**a. Implement Map-reduce operation using**

**MongoDB.**

**b. Implement MongoDB Queries using**

**aggregation and indexing using MongoDB.**

**Program Code:**

**A**] // Map-Reduce to count the number of courses per student

var mapFunction = function() {

emit(this.name, this.courses.length); // Emit student name and the number of courses

};

var reduceFunction = function(key, values) {

return Array.sum(values); // Sum up the course counts for each student

};

db.students.mapReduce(

mapFunction,

reduceFunction,

{ out: "course\_count" } // Output collection

);

// View the result of Map-Reduce

db.course\_count.find();

**B]**// Create a simple index to improve query performance on the 'name' field

db.students.createIndex({ name: 1 });

// Create a compound index on 'name' and 'age' fields for optimized searching

db.students.createIndex({ name: 1, age: 1 });

// Aggregation Query: Count the number of courses for each student

db.students.aggregate([

{

$project: {

name: 1, // Include student name

courseCount: { $size: "$courses" } // Count the number of courses

}

}

]);

// Aggregation Query with filtering: Find students older than 20 and count their courses

db.students.aggregate([

{ $match: { age: { $gt: 20 } } }, // Filter students older than 20

{

$project: {

name: 1,

courseCount: { $size: "$courses" }

}

}

]);

**Output:**

[

// Map-Reduce Result: Course count per student

{ "\_id": "Amit", "value": 3 }, // Amit has 3 courses

{ "\_id": "Priya", "value": 2 }, // Priya has 2 courses

// Index Creation (on 'name' field)

{ "acknowledged": true, "operationTime": ObjectId("...") },

// Compound Index Creation (on 'name' and 'age' fields)

{ "acknowledged": true, "operationTime": ObjectId("...") },

// Aggregation Result: Count the number of courses for each student

{ "\_id": "Amit", "courseCount": 3 }, // Amit has 3 courses

{ "\_id": "Priya", "courseCount": 2 }, // Priya has 2 courses

// Aggregation Result with Filter (Students older than 20)

{ "\_id": "Priya", "courseCount": 2 } // Priya is older than 20 and has 2 courses

]