Write mapreduce function on **students info** collection to find the number of students who scored more than 70 marks in DBMS as " Highscores",less than 70 marks but greater than 40 marks as "Average scores" and less than 40 marks as "Failed

Step 1: Define the Map Function

The map function will emit different categories based on the Marks obtained by each student in the DBMS subject.

Step 2: Define the Reduce Function

The reduce function will aggregate the counts emitted by the map function.

Step 3: Execute the MapReduce Query

Here's how to implement this:

javascript

Copy code

```
// Define the map function
var mapFunction = function() {
  if (this.Subject === "DBMS") {
    if (this.Marks > 70) {
      emit("Highscores", 1);
    } else if (this.Marks > 40) {
      emit("Average scores", 1);
    } else {
      emit("Failed", 1);
    }
  }
};

// Define the reduce function
var reduceFunction = function(key, values) {
  return Array.sum(values);
};

// Execute the MapReduce operation
db.student.mapReduce(
```

```
 mapFunction,
 reduceFunction,
 { out: "student_scores_category" } // This will store the results in a new collection
);
```

Explanation:

1. **Map Function**:
    - Checks if the subject is "DBMS".
    - Emits a key-value pair based on the Marks:
        - "Highscores" with a value of 1 for scores greater than 70.
        - "Average scores" with a value of 1 for scores between 41 and 70.
        - "Failed" with a value of 1 for scores less than or equal to 40.
2. **Reduce Function**:
    - Takes the emitted key and an array of values (counts) and sums them to get the total count for each category.
3. **MapReduce Execution**:
    - Executes the mapReduce function on the student collection.
    - Outputs the result into a new collection called student_scores_category.

Result Verification

To see the results, you can run:

javascript

Copy code

```
db.student_scores_category.find();
```

ANS =