|  |
| --- |
|  |

**LAB 6: PERFORMANCE MONITORING AND OPTIMIZATION**

|  |  |
| --- | --- |
| **Track** | Data & AI |
| **Role** | Software Development - Backend |
| **Level** | Level 2 |
| **Course** | NoSQL Databases |
| **Scope** | * Monitoring MongoDB Performance * Query Optimization Techniques * Using MongoDB Compass for Real-Time Monitoring * Advanced Query Optimization |
| **Prerequisite** | * Understanding of MongoDB indexing and query execution concepts. * Familiarity with MongoDB Compass. * A working MongoDB environment with data. |

**Objective**

To monitor database performance and optimize queries for better execution efficiency and resource utilization.

**Tasks**

**Dataset (also available in resources): telecom\_dataset\_v2.json**

1. Query Analysis Using explain()
2. Implement Indexing for Query Optimization
3. Implement Compound Index for Complex Queries
4. Monitor Query Performance Over Time
5. Use Projections for Efficient Queries
6. Pagination for High Data Volume
7. Analyze Average Monthly Bill by Region
8. Find Customers with Data Usage Above Threshold
9. Measure the Impact of Projections on Query Performance
10. Test Pagination Performance with Indexing

**Outcomes**

* Monitored MongoDB performance metrics using Compass.
* Identified and resolved performance issues through query optimization.
* Achieved faster query execution using indexing, projections, and pagination techniques.
* Reorganized and monitored indexes for better performance.

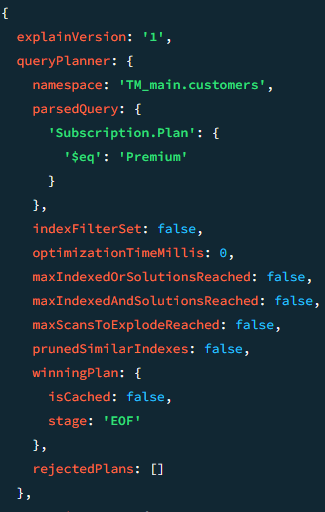
**Solution**

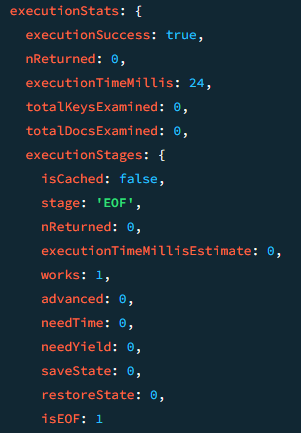
1. **Query Analysis Using explain():**

* Analyzing Query Performance for Premium Customers:
  + Analyze query performance for retrieving customers with the "Premium" subscription plan.



* + Output:



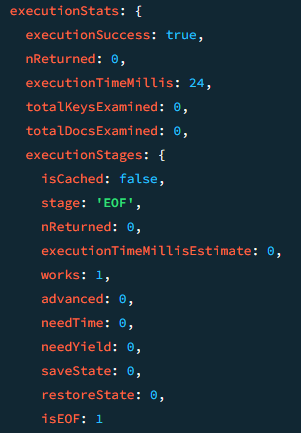


1. **Implement Indexing for Query Optimization:**

* Create an index on the Subscription.Plan field to improve query performance:
  + Create an index on the Subscription.Plan field to improve query performance:



* Before Index:



* Before Optimization:
* No index on Subscription.Plan.
* Collection scan for every query.
* High latency and resource usage.
* Poor scalability for large datasets.
* Re-run the same query and analyze the improvement. First Create Index in Subscription.Plan.





* After Implementing Indexing:
  + Output:

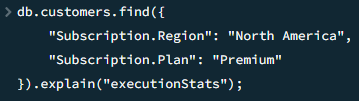


1. **Implement Compound Index for Complex Queries:**

* Compound Index for Complex Queries :
  + Create a compound index on Subscription.Region and Subscription.Plan.

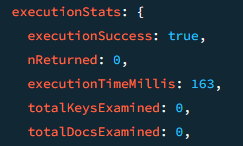


* + Run a query filtering by both fields.



* + Output:





1. **Monitor Query Performance Over Time:**

* Track query execution times to identify patterns and optimize long-running queries:
  + Use the MongoDB Profiler to monitor and log queries that take longer than 50 milliseconds.



* + Output:



* Run a sample query: 
  + Output:

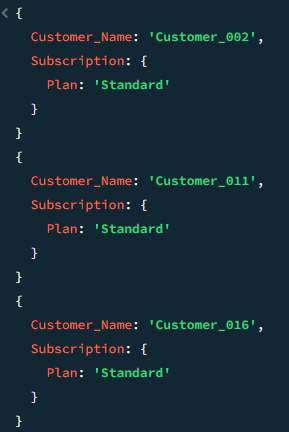


1. **Use Projections for Efficient Queries:**

* Use projections to return only the necessary fields, minimizing data transfer and improving query efficiency.



* + Output:



1. **Pagination for High Data Volume:**

* Use the skip() and limit() functions in the query to fetch specific pages of data, reducing the data load for high-volume datasets.

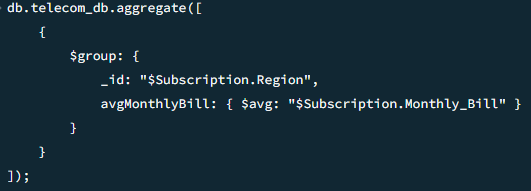


* + Output:



1. **Analyze Average Monthly Bill by Region:**

* Determine the average monthly bill for each region:
  + Run an aggregation query to group customers by region and calculate the average Monthly\_Bill.



* + Output:

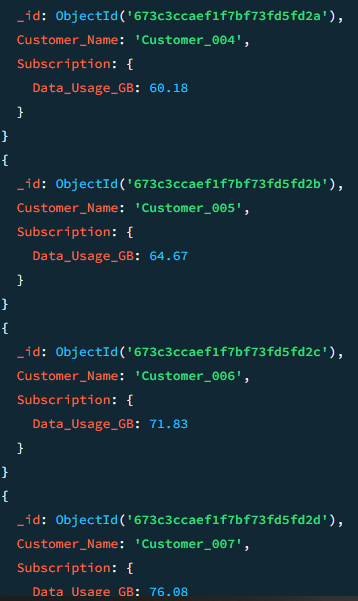


1. **Find Customers with Data Usage Above Threshold:**

* Identify customers with data usage greater than 60GB:
  + Find customers with data usage greater than 60GB



* + Output:

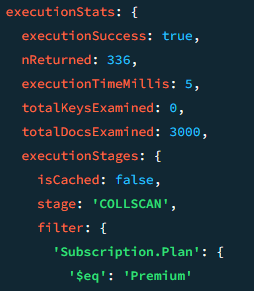


1. **Measure the Impact of Projections on Query Performance:**

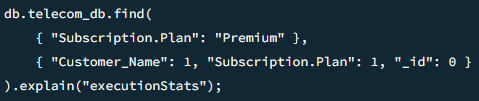
* Optimize query performance by retrieving only required fields:
  + Run a query with and without projections to compare their execution stats.
* Without Projection:



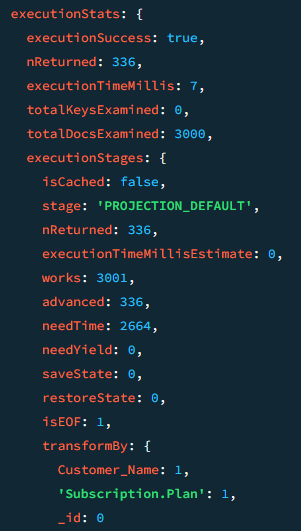
* + Output:



* With Projection:



* + Output:

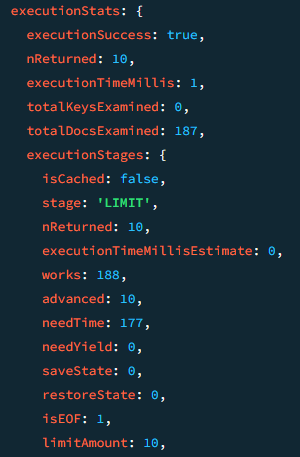


1. **Test Pagination Performance with Indexing:**

* Evaluate the efficiency of pagination queries with indexed fields:
  + Use the limit() and skip() methods to paginate data and compare performance with and without indexing.
* Without Index:



* + Output:



* Add Index:

****

* With Index:



* + Output:

