Course Overview

View Discussion

Chapter 2: MongoDB Indexes

Lab 2.2: Optimizing Compound Indexes

‹ Back to the Question

The key to this lab is to determine which index will provide the most index prefixes that can be utilized by the 3 example queries.

Let's analyze each option:

- `{ "address.state": 1, "job": 1 }`

  No, while this index would be able to service all 3 of the example queries, there's a better index that can be used on the first query, and the second query has to do an in-memory sort.

- `{ "address.state": 1, "job": 1, "first_name": 1 }`

  No, this index is better than the first, but it still doesn't help with the sort on the second query.

- `{ "address.state": 1, "last_name": 1, "job": 1 }`

  Yes, this is the best index. This index matches the first query, can be used for sorting on the second, and has an prefix for the 3rd query.

- `{ "job": 1, "address.state": 1 }`

  No, this index can only be used by the first two queries.

- `{ "job": 1, "address.state": 1, "first_name": 1 }`

  No, while this index is better than the one directly above it, this index still cannot be used by the 3rd query at all.

- `{ "job": 1, "address.state": 1, "last_name": 1 }`

  No, this index has the same issues as the index directly above it.

Proceed to next section