

MongoDB - Index

Topics: MongoDB

Written on December 30, 2023

In MongoDB, an index is a data structure that improves the speed of data retrieval operations on a collection. Indexes can be created on one or more fields of a collection and are used to quickly locate documents that match query criteria.

Creating an Index:

You can create an index using the createIndex method. For example, to create an ascending index on the field name field:

db.collection_name.createIndex({ field_name: 1 });

The 1 specifies ascending order, and -1 would specify descending order.

Compound Indexes:

MongoDB supports compound indexes, where an index is created on multiple fields. For example, to create a compound index on two fields:

db.collection name.createIndex({ field1: 1, field2: -1 });

Index Types:

1. Single Field Index:

An index on a single field.

2. Compound Index:

• An index on multiple fields.

3. Text Index:

Supports text search on string content.

4. Geospatial Index:

• Supports queries that calculate geometries on an earth-like sphere.

5. Hashed Index:

• Useful for equality queries but not range queries.

Using an Index:

MongoDB automatically uses an index to optimize queries when the query criteria match the index. You can use the explain method to see if a query is using an index:

db.collection name.find({ field name: value }).explain("executionStats");

Deleting an Index:

You can delete an index using the dropIndex method:

db.collection name.dropIndex({ field name: 1 });

Indexing Best Practices:

1. Analyze Query Patterns:

• Analyze the queries your application performs to determine which fields to index.

2. Consider Compound Indexes:

• For queries that involve multiple fields, consider creating compound indexes.

3. Avoid Over-Indexing:

• While indexes improve read performance, they come with a cost during write operations. Avoid creating too many indexes.

4. Use Indexes for Sort Operations:

• If your queries involve sorting, consider creating an index on the sorted field.

5. Monitor Index Performance:

 \circ Use tools like the MongoDB Database Profiler or the explain method to monitor and optimize query performance.

6. Use Covered Queries:

 Create indexes that cover the queries so that the index itself provides the necessary data.

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by <u>Aryatechno</u>