|  |  |
| --- | --- |
| MongoDB Questions |  |
|
|

1. **What is the primary purpose of an index in MongoDB?**

* The primary purpose of an index in MongoDB is to improve the speed of queries by efficiently locating documents in a collection.

1. **Name and brieﬂy describe types of indexes available in MongoDB?**

**Types of indexes in MongoDB:**

* Single Field Index: Indexes a single field in a collection.
* Compound Index: Indexes multiple fields in a collection.
* Multikey Index: Indexes arrays within documents.
* Geospatial Index: Indexes geospatial coordinate data for location-based queries.
* Text Index: Indexes text data for text search.
* Hashed Index: Indexes a field's hashed value for sharding.

1. **How would you create a compound index on the ﬁelds ﬁrstName and lastName in a collection called users?**

* db.users.createIndex({ firstName: 1, lastName: 1 })

1. **What are some potential downsides to creating too many indexes on a collection?**

* Increased storage requirements: Indexes consume additional disk space.
* Slower write operations: Indexes need to be updated on every write, insert, update, or delete operation.
* Complexity in management: More indexes can make maintenance and optimization more complicated.

1. **How do you create a unique index on the email ﬁeld of a customer’s collection?**

* db.customers.createIndex({ email: 1 }, { unique: true })

1. **What is the purpose of the aggregation framework in MongoDB?**

* The purpose of the aggregation framework in MongoDB is to process and transform data by performing operations such as filtering, grouping, and sorting, to generate computed results from a collection.

1. **List the basic stages of an aggregation pipeline and brieﬂy describe their functions.**

* $match: Filters documents to pass only those that match specified criteria.
* $group: Groups documents by a specified field and applies aggregate functions like sum, avg, min, max.
* $project: Selects specific fields to include or exclude in the output documents, and can add new computed fields.
* $sort: Orders the documents based on the values of specified fields, either ascending or descending.
* $limit: Restricts the number of documents in the output to a specified count.
* $skip: Skips a specified number of documents, useful for pagination.

1. **Write an aggregation pipeline that groups documents in a sales collection by product Id and calculates the total quantity sold for each product.**

* db.sales.aggregate ([

{$group: {\_id: "$productId", totalQuantitySold: {$sum: "$quantity”}}}

])

1. **Explain the purpose of the $lookup stage in the aggregation pipeline.**

* The purpose of the `$lookup` stage in the aggregation pipeline in MongoDB is to perform a left outer join to another collection, combining documents from both collections based on a specified condition.

1. **How would you use the aggregation framework to ﬁnd the average order value from an orders collection?**

* db.orders.aggregate ([

{$group: {\_id: null, avgOrderValue: {$avg: "$totalAmount”} } }

])

1. **How do you create a text index on the description ﬁeld of a products collection?**

* db.products.createIndex({ description: "text" })

1. **What is the purpose of the $facet stage in an aggregation pipeline?**

* The purpose of the `$facet` stage in an aggregation pipeline in MongoDB is to compute multiple independent aggregations on the same set of input documents, producing multiple sets of results within a single aggregation operation**.**

1. **What are some best practices for optimizing aggregation pipelines in MongoDB?**

* Use Indexes.
* Limit Data.
* Project Only Needed Fields.
* Optimize `$group` Operations.
* Use `$lookup` wisely.
* Avoid `$unwind` If Possible.
* Pipeline Reordering.
* Avoid large `allowDiskUse’.

1. **When would you use the Map-Reduce framework instead of the aggregation framework in MongoDB?**

* Can use the Map-Reduce framework in MongoDB when we need to perform complex data transformations and aggregations that cannot be easily achieved using the aggregation framework, especially for operations involving extensive computation or processing of large datasets.