Experiment: 10

Develop an aggregation pipeline to illustrate Text search on Catalog data collection

An aggregation pipeline is a sequence of stages that process documents and return computed results. Each stage in the pipeline performs an operation on the documents, and the output of one stage becomes the input for the next.

Common Stages:

Here are some of the most commonly used stages in an aggregation pipeline:

1.$match: Filters documents based on specified conditions.

{ $match: { field: value } }

2.$project: Selects or excludes fields from the documents.

{ $project: { field1: 1, field2: 0 } }

3.$group: Groups documents by specified fields and calculates aggregations.

{ $group: { _id: "$field", total: { $sum: "$value" } } }

4.$sort: Sorts documents based on specified fields.

{ $sort: { field: 1 } } // 1 for ascending,-1 for descending

5. $limit: Limits the number of documents returned.

{ $limit: 10 }

6.$skip: Skips a specified number of documents.

{ $skip: 20 }

7. $unwind: Decomposes arrays into separate documents.

{ $unwind: "$arrayField" }

8. $lookup: Joins data from other collections.

{ $lookup: { from: "collection2", localField: "field1", foreignField: "field2", as: "results" } }

```
db.catalog_data.aggregate([
// Stage 1: Match documents containing the search term
{
$match: {
$text: {
$search: "search_term"
}
}
},
// Stage 2: Project only required fields
{
$project: {
_id: 0,
productName: 1,
description: 1,
// Add more fields as needed
}
}
])
```

EXPLANATION:

Stage 1: $match

- **db.catalog_data.aggregate([**: This line initiates an aggregation pipeline on the catalog_data collection.
- **$match: { $text: { $search: "search_term" } }**: This stage filters documents based on the specified search term.
    - $match: Specifies the matching criteria.
    - $text: Indicates a text search.
    - $search: Defines the search term. Replace "search_term" with the actual search query.

Stage 2: $project

- **$project: { ... }**: This stage selects and transforms the documents.

   - _id: 0: Excludes the _id field from the output.
   - productName: 1: Includes the productName field in the output.
   - description: 1: Includes the description field in the output.
   - You can add more fields as needed to include them in the output.

Overall, this aggregation pipeline:

1. Searches the catalog_data collection for documents containing the specified search term.
2. Selects only the productName and description fields from the matching documents.
3. Returns the results as an array of documents.