Search Pipeline use cases

Below is a list of search pipelines enabling you to alternate the results of a search process based on the desired capability. You can use COMPASS ‘new pipeline from text’ functionality to copy and paste pipeline code and see the results.

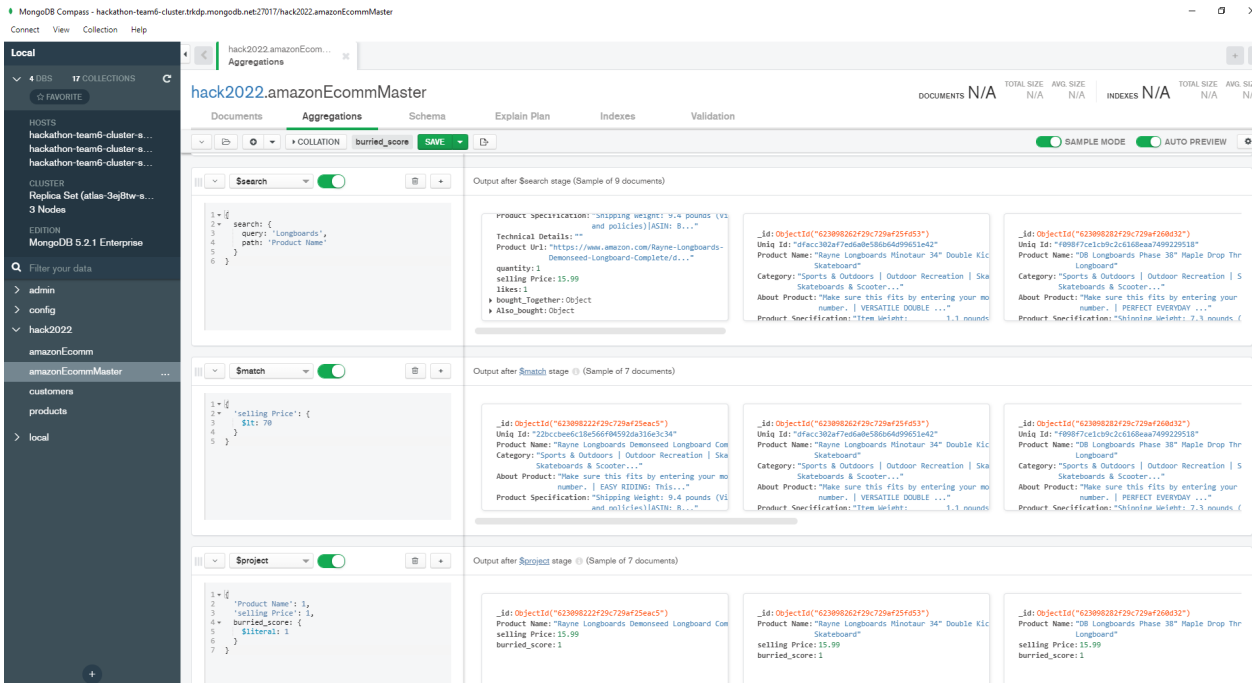
The Bury, Boost, Pin, Hide all uses the default dynamic index construct of the form:

```
{
  "mappings": {
    "dynamic": true
  }
}
```

The last Pipeline - diacritics abstraction - uses a custom analyser definition.

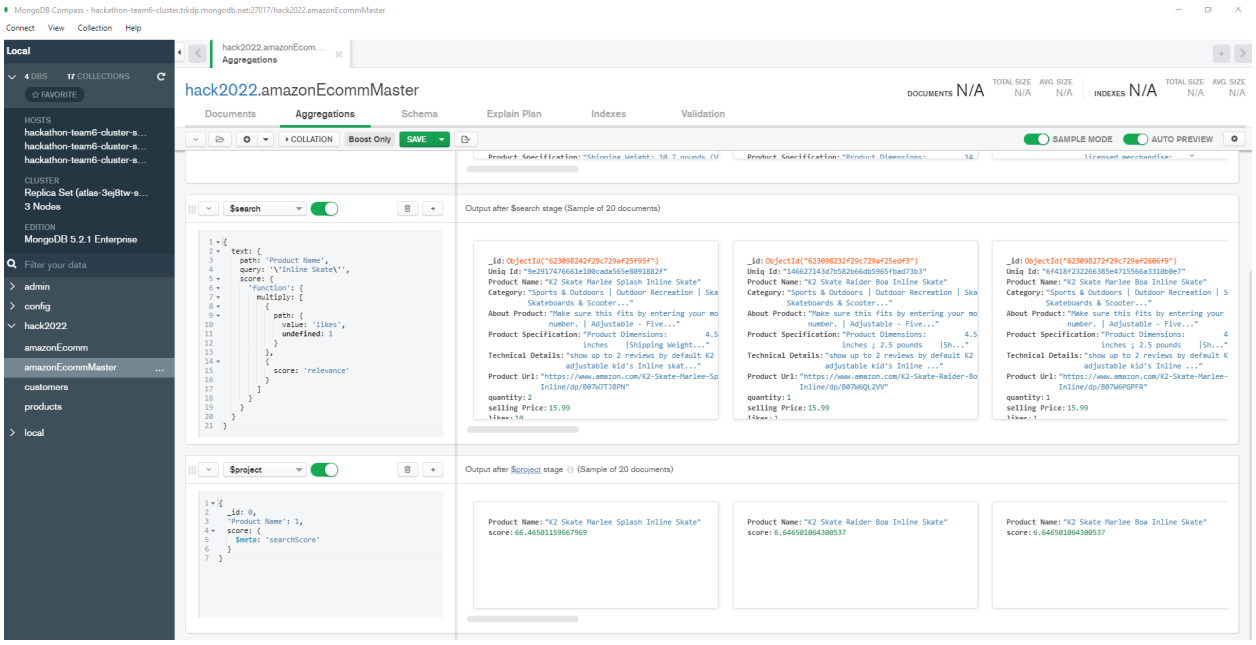
Bury

Bury pipeline function can be achieved by matching the desired documents you want to bury and override the Lucine relevancy score with a constant value. Application logic leverages this constant score to process documents.Hence all matching documents will yield a uniform score effectively burring these documents amongst the entire dataset.



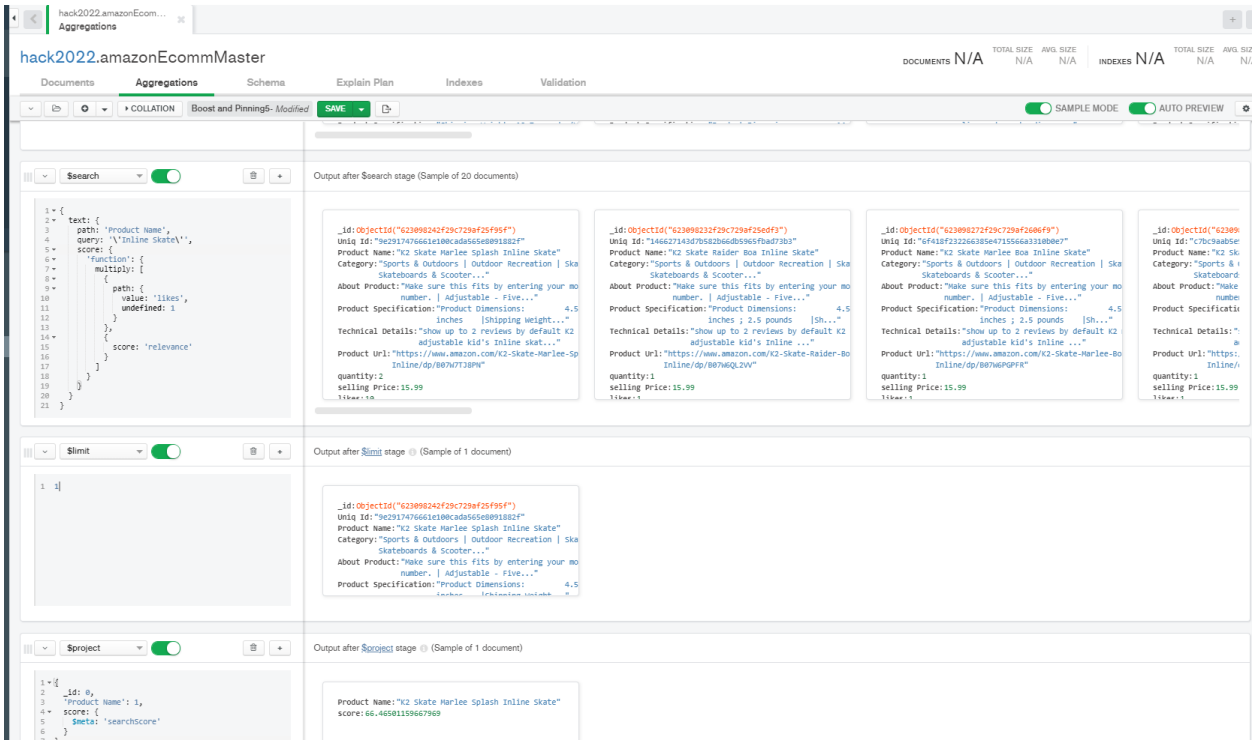
Boost

The boost function enables you to augment the original Lucine relevance score enabling you to influence the overall score by using additional factors. For instance within the context of an Ecommerce site, the ability to control the order of presentation of various articles for sales (ie documents) can be a function of the native relevance score from Lucine multiplied by the number of likes associated with each product items.



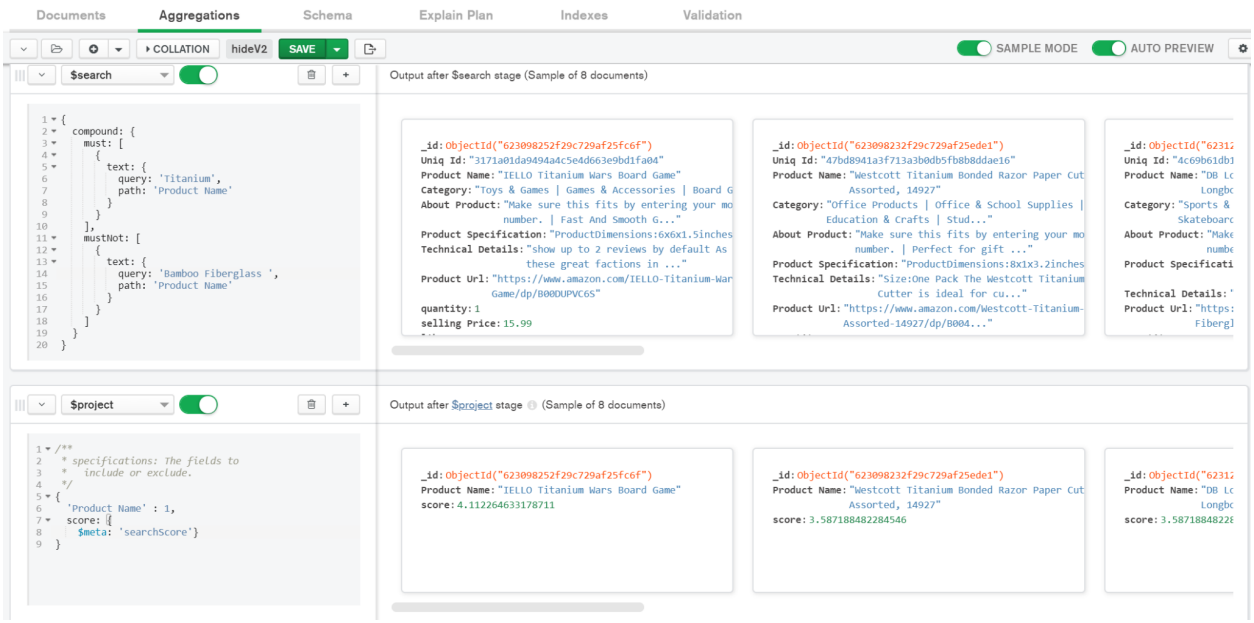
Pin

Pinning of documents can be implemented by controlling the quantity of documents returned by the pipeline. This is easily done by defining a limit stage within your aggregation pipeline. The Pin function can be utilized in addition to other functions - like boost. You can control the number of documents to Pin just by changing the limit criteria - easy…



Hide

Hide capability can be implemented by using a compound search function. In the pipeline below we are searching for products that are made of Titanium and we are excluding products that are made in bamboo or Fiberglass.This can be accomplished using a single search stage within Aggregation Pipeline, There is no need to add any additional attributes or business logic to implement this function.



Diacritics Abstraction

This capability applies to ECommerce websites that service global customers, where various locals use diacritic symbols. For instance in French a word can be spelled out correctly using various representations (for instance the first name benoit can also be spelled benoît). The custom analyser will deliver documents which contain both representations of benoit (and benoît) without any regards to the actual encoding.

By implementing a custom analyser we can abstract the diacritic representation, delivering a superior user experience.

Using Mongo shell let's query the amazonEcommMaster

Using the default analyser - noticed that the query only returns a single document with the author name = benoit

```
MongoDB Enterprise atlas-3ej8tw-shard-0:PRIMARY> db.amazonEcommMaster.aggregate([
...   {
...     $search: {
...       "index": "default",
...       "text": {
...         "query": "benoît",
...         "path": "author"
...       }
...     }
...   },
...   {
...     $project: {
...       "_id": 1,
...       "author": 1
...     }
...   }
... ])
{ "_id" : ObjectId("6230e6a892cdccde2e4a549f"), "author" : "benoît" }
MongoDB Enterprise atlas-3ej8tw-shard-0:PRIMARY>
```

Using the custom analyser - asciiConverter, noticed the engine returns both representations benoit (and benoît) regardless of the search query parameter thus abstracting the diacritic representation of the attribute author.

```
... ]])
MongoDB Enterprise atlas-3ej8tw-shard-0:PRIMARY> db.amazonEcommMaster.aggregate([
...   {
...     $search: {
...       "index": "asciiConverter",
...       "text": {
...         "query": "benoît",
...         "path": "author"
...       }
...     }
...   },
...   {
...     $project: {
...       "_id": 1,
...       "author": 1
...     }
...   }
... ])
{ "_id" : ObjectId("6230e67192cdccde2e4a549e"), "author" : "benoit" }
{ "_id" : ObjectId("6230e6a892cdccde2e4a549f"), "author" : "benoît" }
MongoDB Enterprise atlas-3ej8tw-shard-0:PRIMARY>
```

Hope you enjoy