In MongoDB, indexing refers to the execution of its queries. (docs.mongodb.com) Indexing is incredibly important because without it, the database would have to search with a collection scan which would be incredibly slow in comparison.

Indexes store a special data structure that stores a small portion of the collections data set (2). In other words, the indexes are storing the value of a specific field in a document. This ensures there are exact matches in the querying operations.

One of the default ways MongoDB indexes is the use of the \_id field. MongoDB automatically adds this field so that it isn’t possible to create two of the same collections. Then it is possible to index through the use of queries for these unique indices.

Index cardinality refers to the uniqueness of values stored in a specified column within and index. There can be low or High cardinality with low meaning there are less unique values and high meaning there are more unique values as general rule of thumb. The higher the cardinality the longer it takes to query in most cases a well and vice versa.

Overall a capped collection works similarly to a regular collection, but it is fixed in size. You will declare the size of the collection and the collection will stay within that size as you insert documents etc. Two common ways of capping a collection is to define its storage space or to define how many documents you want within it. If you specify a capped collection at 100 documents, once you add each additional one past the 100, the oldest document is automatically removed. This method is good for keeping collections at certain sizes and it has the added benefit of auto updating by deleting the oldest of the documents as well once a new one is inserted.

Often times in a NoSQL data base there are keys and values that exist in one document but not the next. This can create a problem when there are set of documents that have all the same keys but there are similar documents in the collection without some of those keys and you want to index to a new key that is not in all the documents. A sparse index helps to organize a new index for the documents that contain the new unique key you want to index to. However, this will mean that the documents that do not contain the new required key will be left out of the new sparse index. Traditionally indexes are just mapped to keys that are all within a document and that are unique such as the built in \_id index in MongoDB.

Resources:

<https://docs.mongodb.com/manual/core/capped-collections/#:~:text=Capped%20collections%20are%20fixed%2Dsize,documents%20based%20on%20insertion%20order>.