For this module's discussion board assignment, research “MongoDB indexing” and answer the following questions:

1. What is indexing and when should you use it?
2. What is index cardinality and why is it important?
3. What are capped collections? And, when should you use them?
4. What are the differences between sparse and traditional indexes?

Since collections in MongoDB are just groups of similar documents, it is important to be able to try and give some order to the collection. It effectively allows for increased performance in queries because the index can allow for a certain bit of information to act as an index and the index can remain sorted. MongoDB has a good example of how indexing can improve performance.

Index cardinality describes how unique the values stored in a field of a document are. The higher the cardinality in an index, the more useful it is because it means there are more unique values in the index.

A capped collection is one which has a fixed size. As the maximum number of documents is reached, the oldest are dropped and replaced by newly created ones. Think of it like how a dashcam works – it fills up your SD card until it’s full then begins to overwrite the oldest footage. This can be useful in database situations when you’re storing log files. If an application has been working well for a month, then you don’t really need logs older than that in most cases.

A sparse index will only contain the information that is being indexed, while a tradition index would contain that as well as the rest of the document. A sparse index is generally faster than one that stores the entire document because the database needs to return less information for a given query (since only one field is stored there)