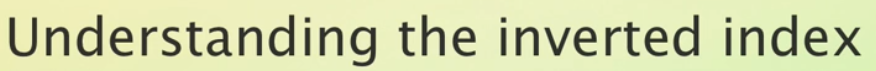
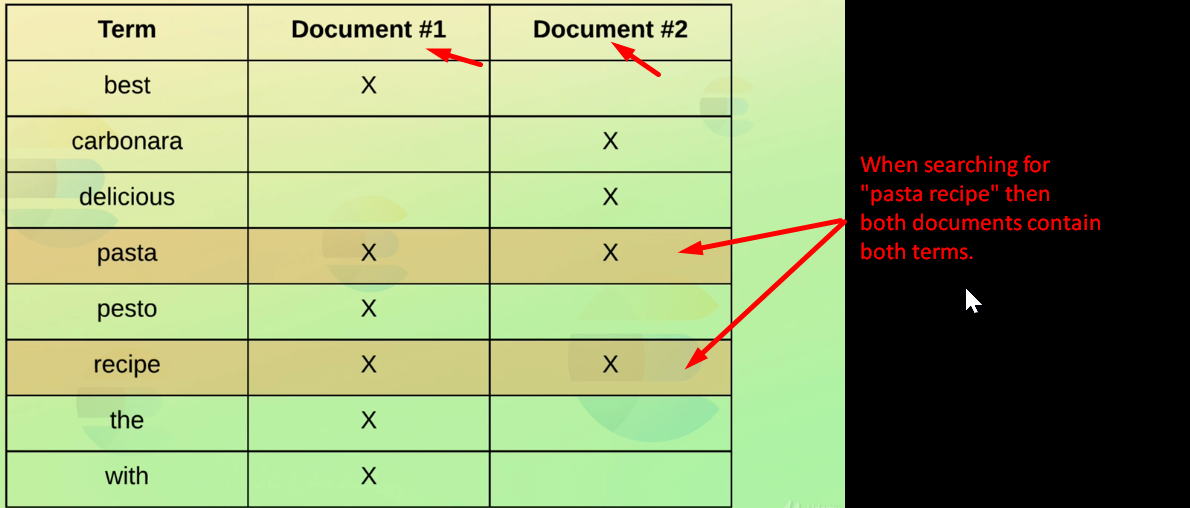
1. 
2. Now that you know how elasticsearch **analyzes text fields**, you might wonder what actually happens with the results of the **analysis process**.
3. They must end up being stored somewhere right because otherwise what’s the point.
4. So, the results are stored within something called **“inverted index”**.
5. The purpose of an **inverted index** is to store text in a **structure** that allows for very efficient and **fast full text searches**.  
   When performing full **text searches**, **we’re actually querying the inverted index not the indexed document.**
6. There is an inverted index for each full text field per index.  
   So, if you have an index containing documents that contain five full text fields you will have five inverted indices. 🡸 How many inverted indices.  
   An inverted index consists of all of the unique terms that appear in any document covered by the index.
7. For each term, there is document in which the term appears is stored.
8. **Inverted Index**: An inverted index is a mapping b/w terms and which documents contain those **terms.**
9. Since an inverted index works at the document field level and stores the terms of a given field. It doesn’t need to deal with different fields.  
   So what you will see in the following example is that the scope of a specific field.
10. So, let’s see an example.
    1. Suppose that we have two recipes with the following titles.  
       🡺 “The best pasta recipe with pesto”   
       🡺 “Delicious pasta carbonara recipe.
    2. The following table shows what the inverted index would look like.  
       
    3. So, the terms from both of the titles have been added to the index
    4. For each term, we can see which document contains the term which enables elasticsearch to efficiently match documents containing specific terms.
    5. A part of what makes this (efficiency) possible is that the terms are sorted.
    6. Also notice that the terms in the index are the results of the analysis process that you saw in the previous lectures.  
       So, most symbols have been removed at this point and characters have been lowercase.  
       This of course depends on the analyzer that was used but that will most often be the standard analyzer.
    7. Performing a search involves a lot of things such as **relevance** which we will discuss later.
    8. The first step of a search query is to find the documents that match the query in the first place.
    9. **For example**: If we search for “pasta recipe”, then we would see that both documents contain both terms.  
       
    10. 
11. Inverted index also holds other information that is used internally such as for **computing relevance**.  
    Some examples of this could be the number of documents containing each term.  
    The number of times a term appears in a given document, the average length of a field etc.  
    As I mentioned in one of the previous lectures, it is also possible to apply stemming of words and synonyms. This would also be applied to the inverted index but I wanted to keep it simple.  
    We will see both stemming and synonyms later in the course.
12. **Brief Recap**:
    1. Analyzer is applied to full text fields.
    2. The result of this analysis process is stored within an **inverted index**.
    3. **Inverted index** consists of all of the terms for a given field across all documents within an index.
    4. Search query is applied to inverted index.   
       This is important to understand as otherwise you will be puzzled if your search query doesn’t match what you expect.