# Understanding the Keyword Data Type in Elasticsearch

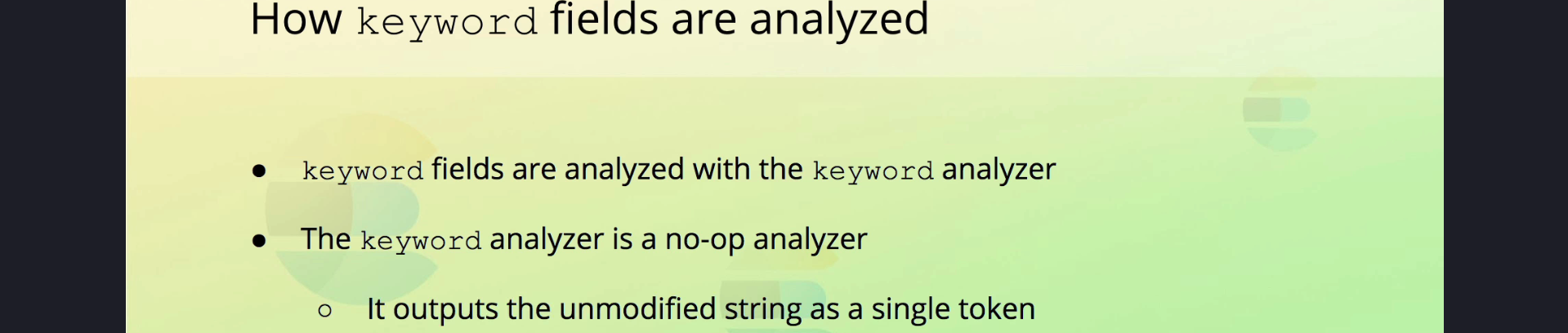
## 1. Introduction

In this lecture, we delve into the purpose and limitations of the 'keyword' data type in Elasticsearch. While it is primarily used for filtering, sorting, and aggregations, it cannot be used for full-text searches. To understand why, we need to explore how 'keyword' fields are analyzed and stored.

## 2. Keyword Analyzer vs. Standard Analyzer

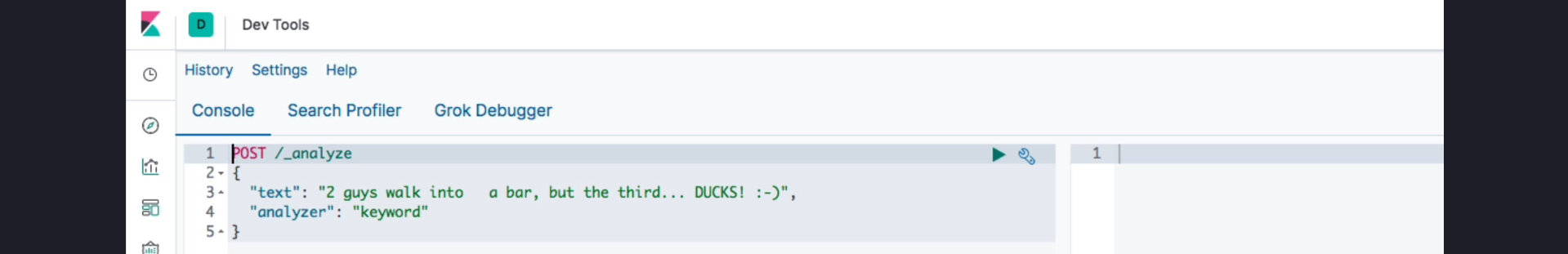
1. The 'keyword' analyzer is used for 'keyword' fields, whereas the 'standard' analyzer is used for 'text' fields.

2. The 'keyword' analyzer is a no-op analyzer, meaning it does not modify the input string. It returns the unmodified string as a single token.

3. This behavior ensures that the values remain intact, enabling exact matches for filtering, sorting, and aggregations.  
  


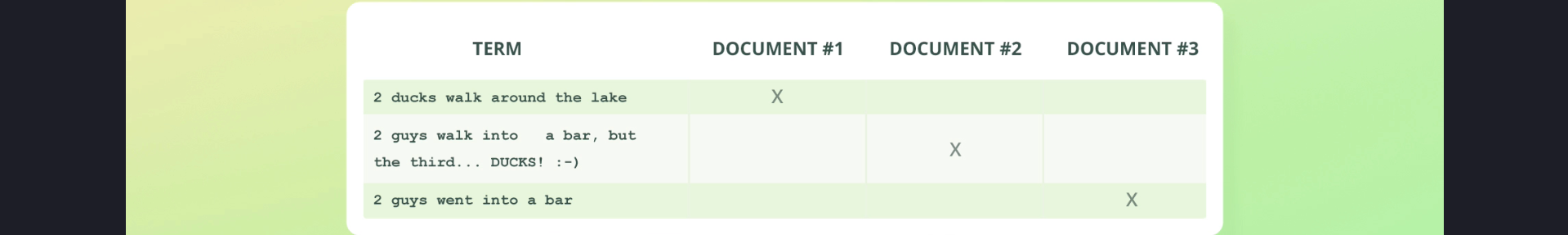
## 3. Verifying Keyword Analyzer Behavior

The behavior of the 'keyword' analyzer can be verified using the Analyze API. Running a query with the API demonstrates that the input string is left entirely untouched and returned as a single token.

Example API Query:  
  
  
**Output**:   


## 4. Inverted Index for Keyword Fields

1. Unlike 'text' fields, the entire field values of 'keyword' fields are stored as single terms in the inverted index.

2. The 'keyword' analyzer does not perform tokenization, remove symbols, or lowercase letters, ensuring exact matches.   
**See below example**:  
  
  
  


## 5. Realistic Use Cases for Keyword Fields

1. Practical use cases for 'keyword' fields include storing data like email addresses, order statuses, and product tags, where exact matches are required.

2. Example Inverted Index:  
{  
 "email": "USER@example.com",  
 "status": "confirmed"  
}

3. Notice that one email address contains uppercase letters. While this may be acceptable for some scenarios, email addresses are not case-sensitive, so using a lowercase token filter could be beneficial.

## 6. Configuring the Keyword Analyzer

1. The 'keyword' analyzer can be configured to include token filters, such as 'lowercase', to modify the behavior.

2. Example Configuration:  
PUT /example\_index  
{  
 "mappings": {  
 "properties": {  
 "email": {  
 "type": "keyword",  
 "normalizer": "lowercase\_normalizer"  
 }  
 }  
 },  
 "settings": {  
 "analysis": {  
 "normalizer": {  
 "lowercase\_normalizer": {  
 "type": "custom",  
 "char\_filter": [],  
 "filter": ["lowercase"]  
 }  
 }  
 }  
 }  
}

## 7. Summary

1. Fields using the 'keyword' data type are analyzed with the 'keyword' analyzer, which outputs the input string as a single unmodified token.

2. The 'keyword' data type is ideal for exact matching, filtering, sorting, and aggregations. Examples include structured data like email addresses and order statuses.

3. The 'keyword' analyzer can be customized using token filters to adapt to specific requirements.  
