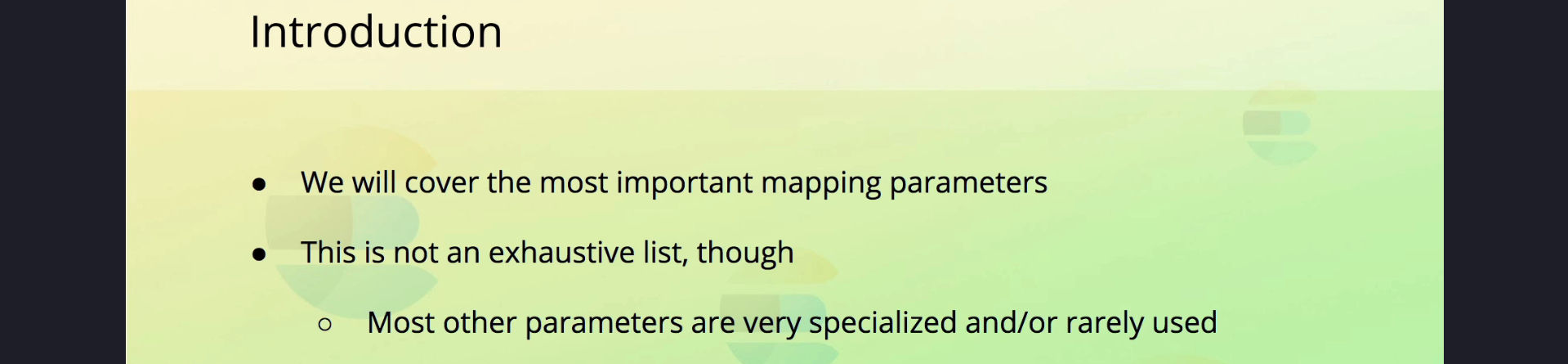
# 

# Overview of Mapping Parameters in Elasticsearch

## Introduction

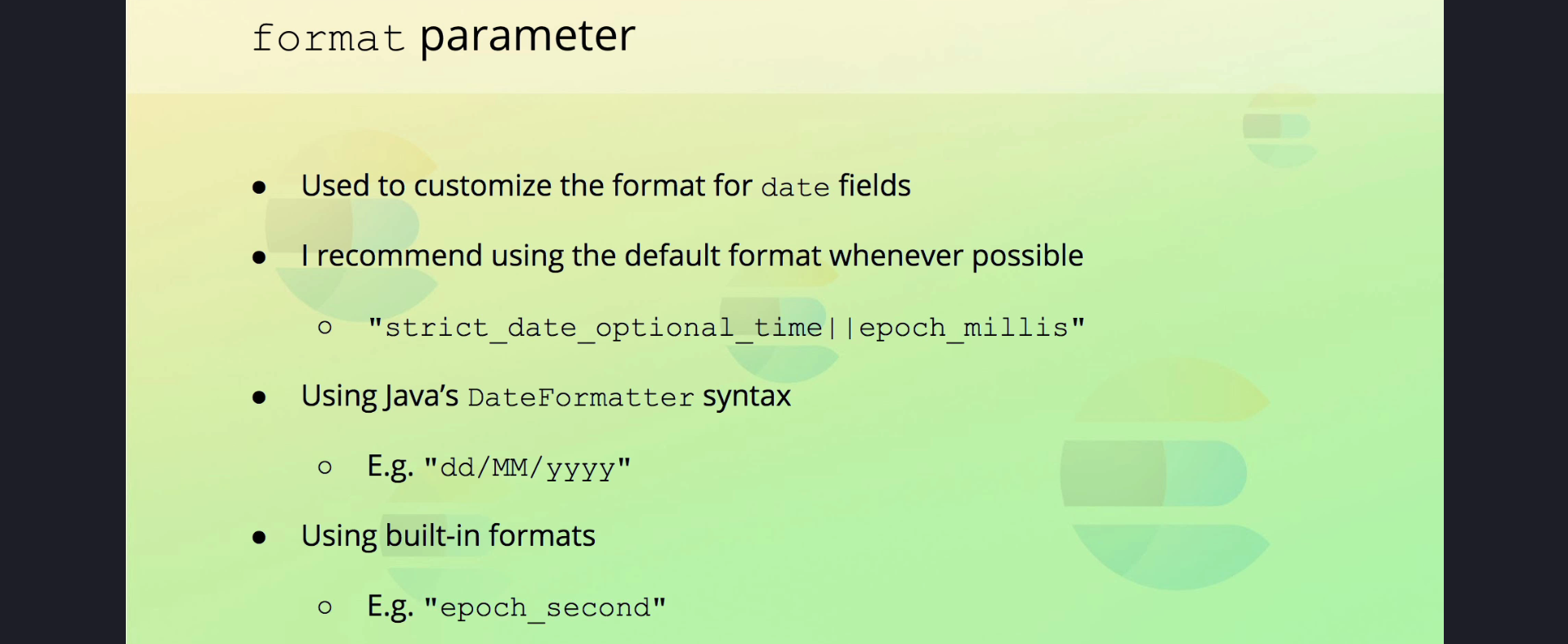
You saw the simple mapping specifying the field name and its data type. Apart from data type, there are other parameters available that configure the behavior of the fields in various ways.

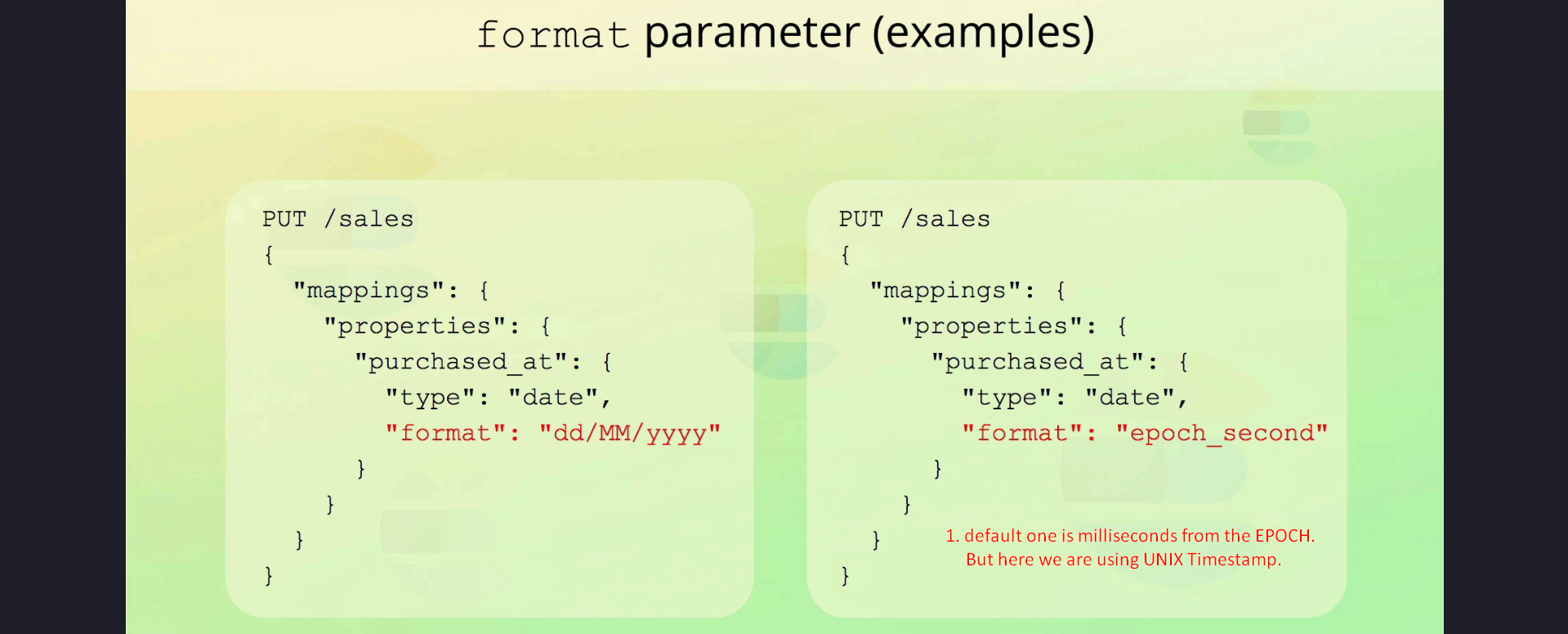
Elasticsearch provides various parameters that configure the behavior of fields in mappings. These parameters help customize the way fields are indexed and searched. This document provides an overview of some of the most important parameters. Note that this is not an exhaustive list.



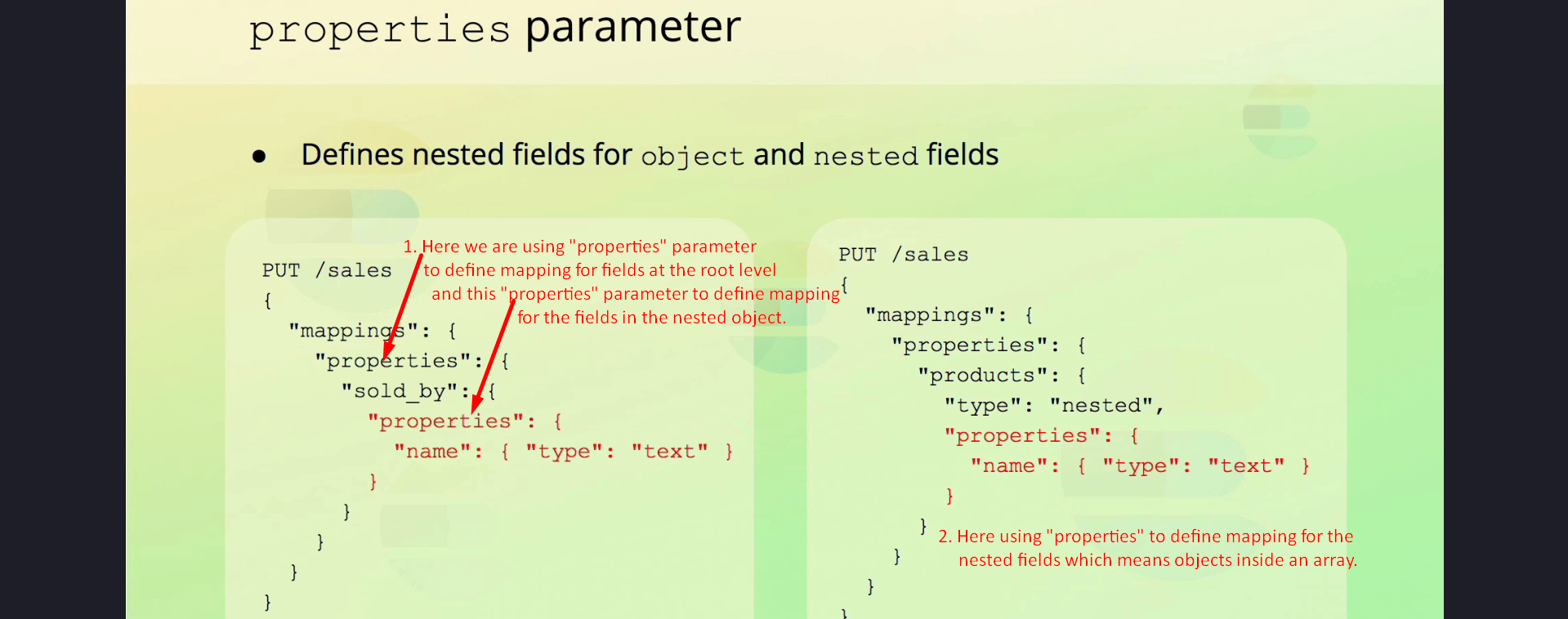
## 2. Format Parameter

The 'format' parameter is used to specify a custom date format when indexing dates that do not conform to the default format. While the default ISO 8601 format is recommended due to its widespread support, custom formats can be defined when necessary, such as when integrating with legacy systems. Elasticsearch supports Java’s DateFormatter class and several built-in formats, like 'epoch\_seconds' for UNIX timestamps.

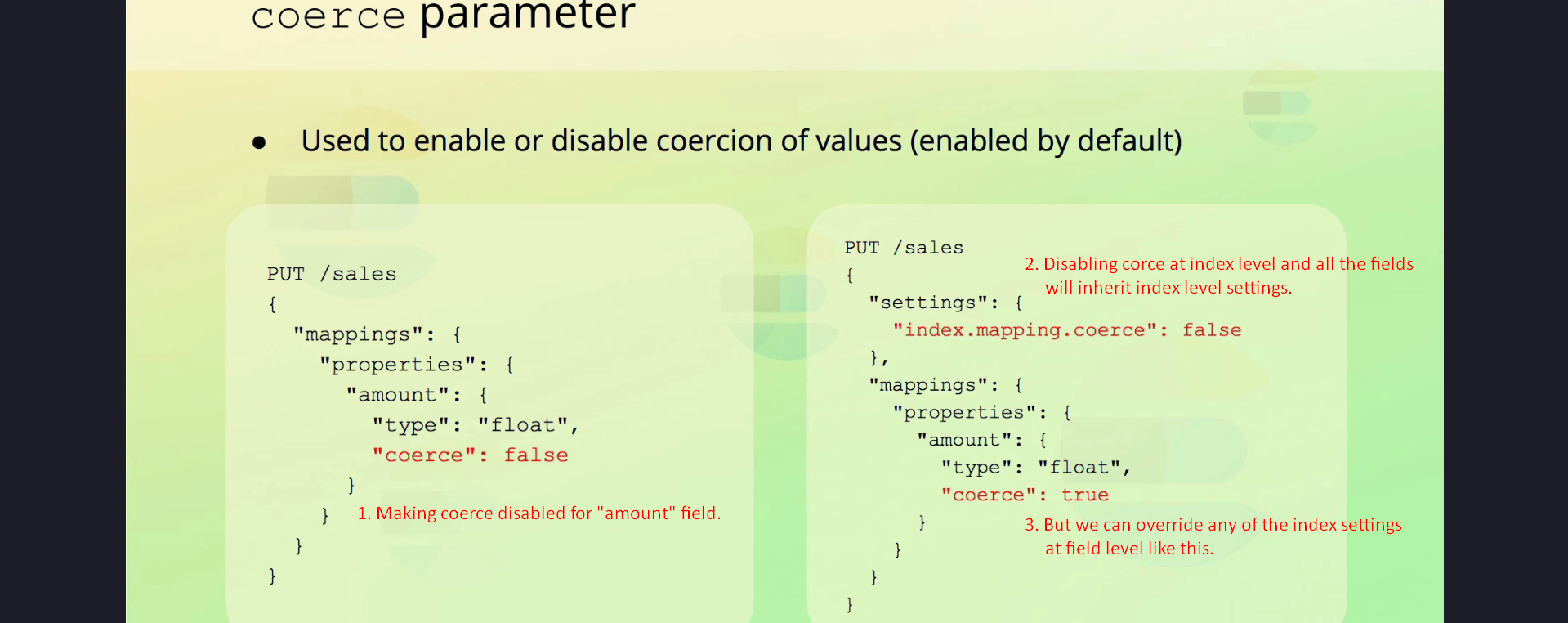




## 3. Properties Parameter

The 'properties' parameter is used to define field mappings at every level of a hierarchy. It is employed for both 'object' and 'nested' data types to map nested fields. While 'object' fields are mapped implicitly using the 'properties' parameter, there is no specific 'object' data type in Elasticsearch.  


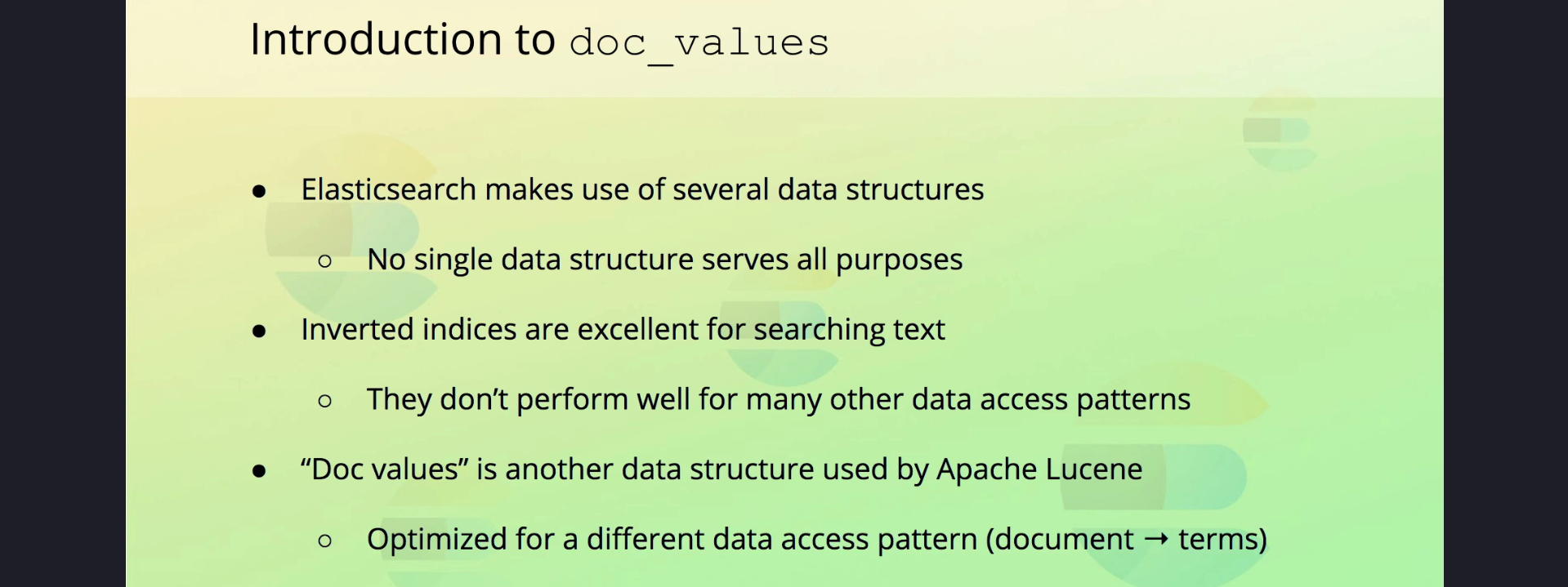
## 4. Coerce Parameter

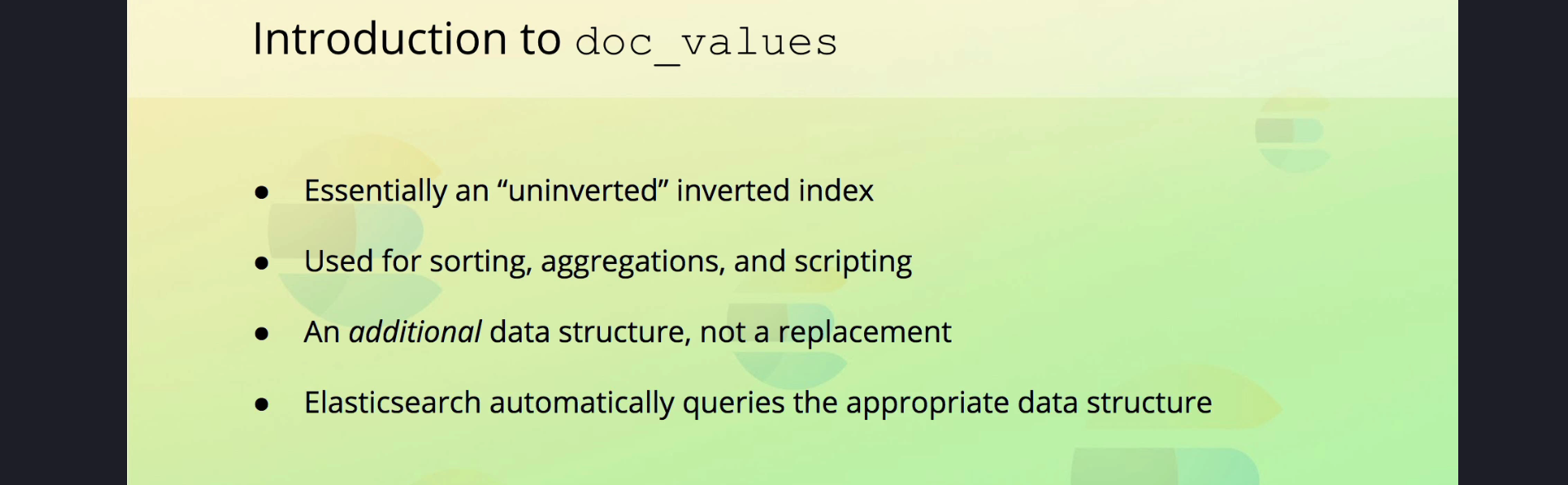
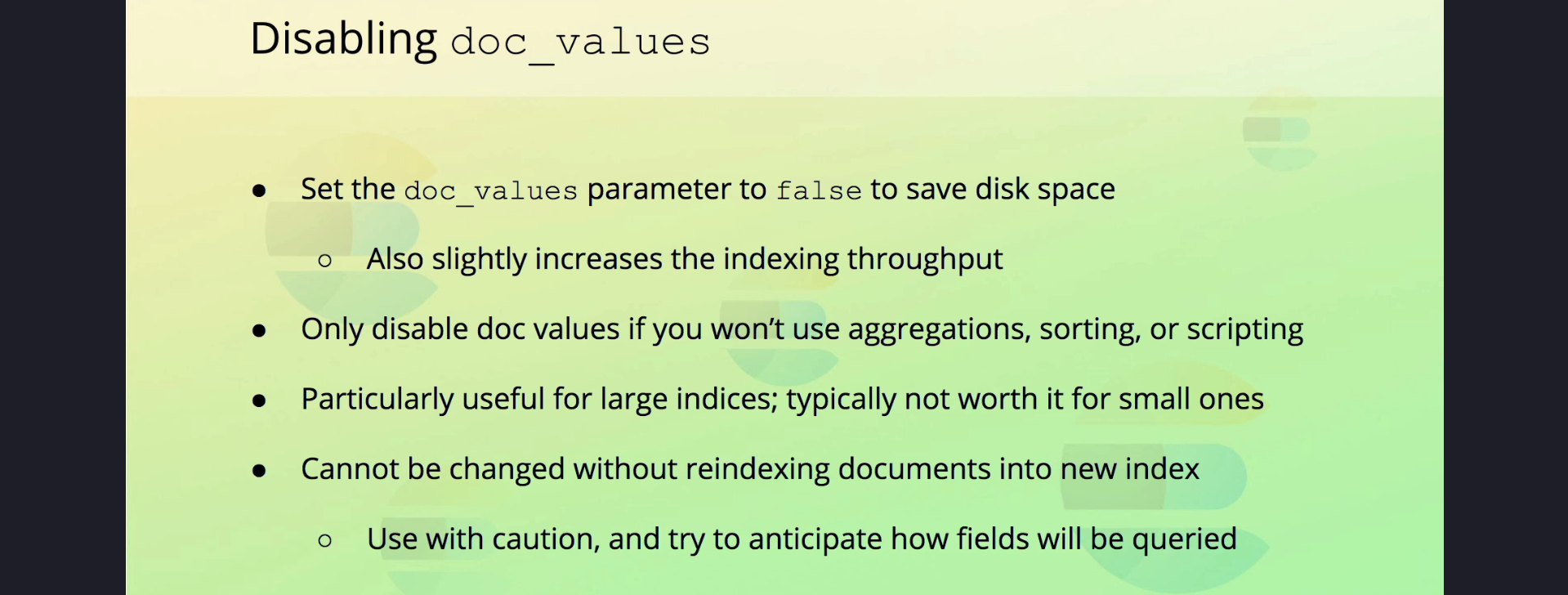
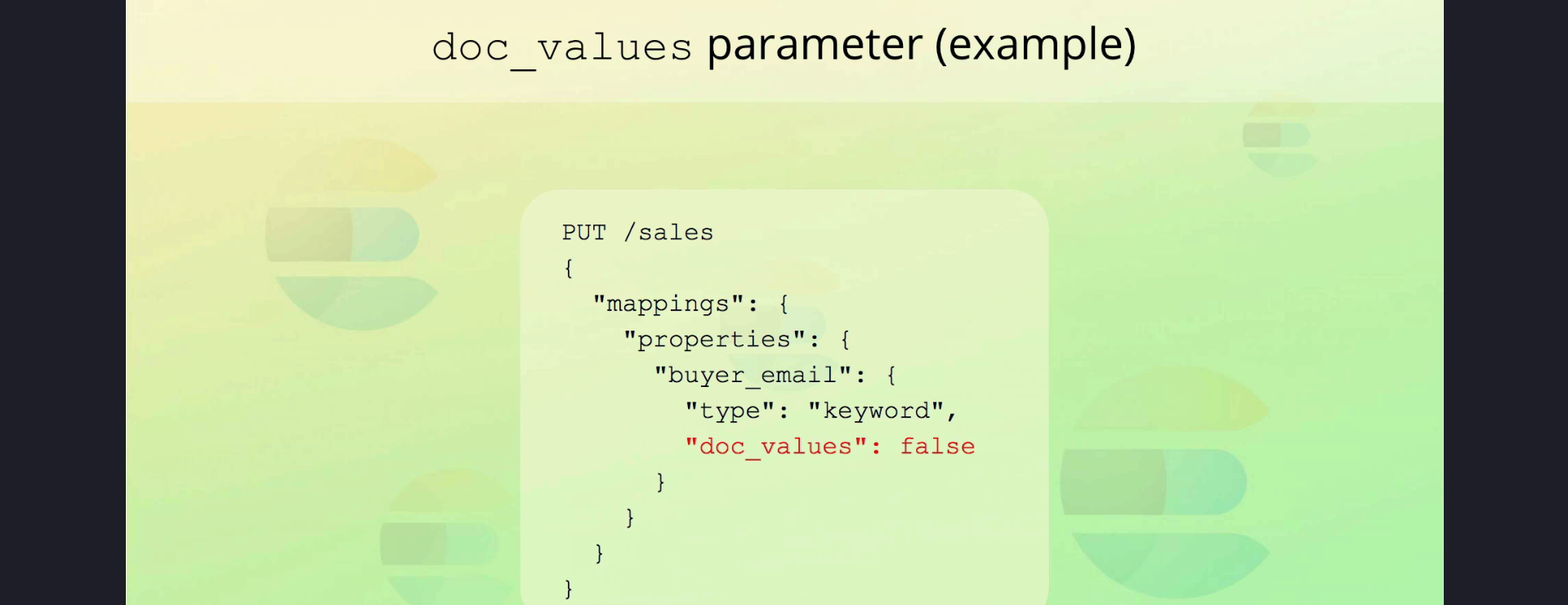


The 'coerce' parameter controls whether type coercion is enabled for a field. Coercion is enabled by default, allowing Elasticsearch to attempt converting values to the appropriate data type. For example, a string representing a numeric value can be coerced into a number. Coercion can be disabled at the index level or for specific fields.

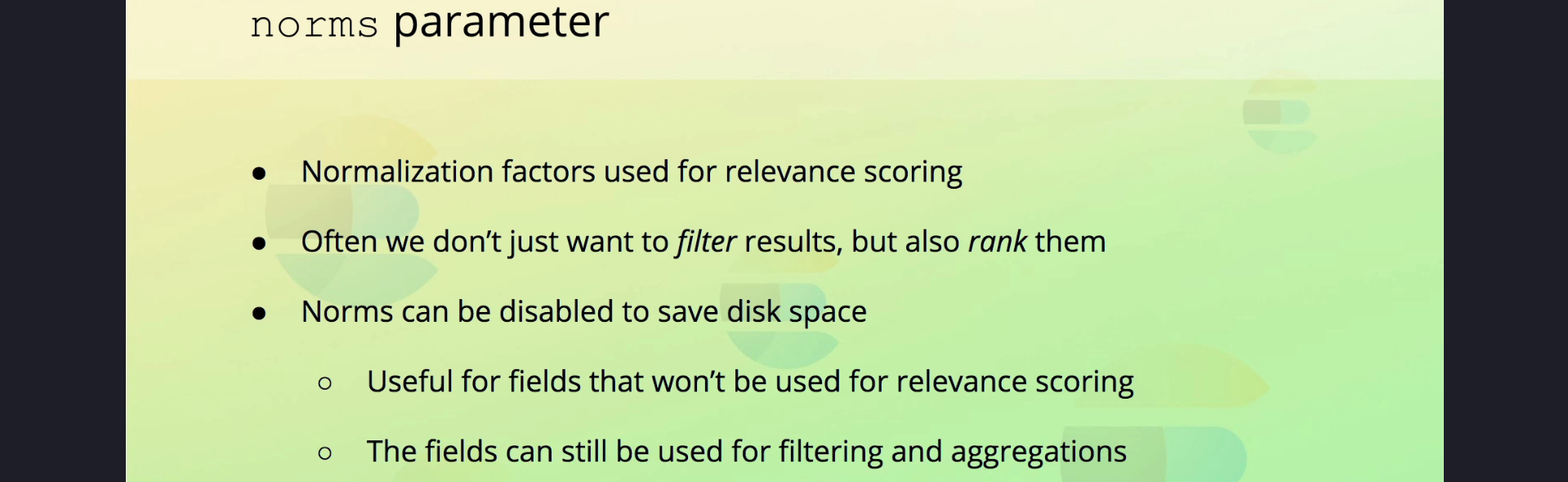
## 5. Doc Values Parameter

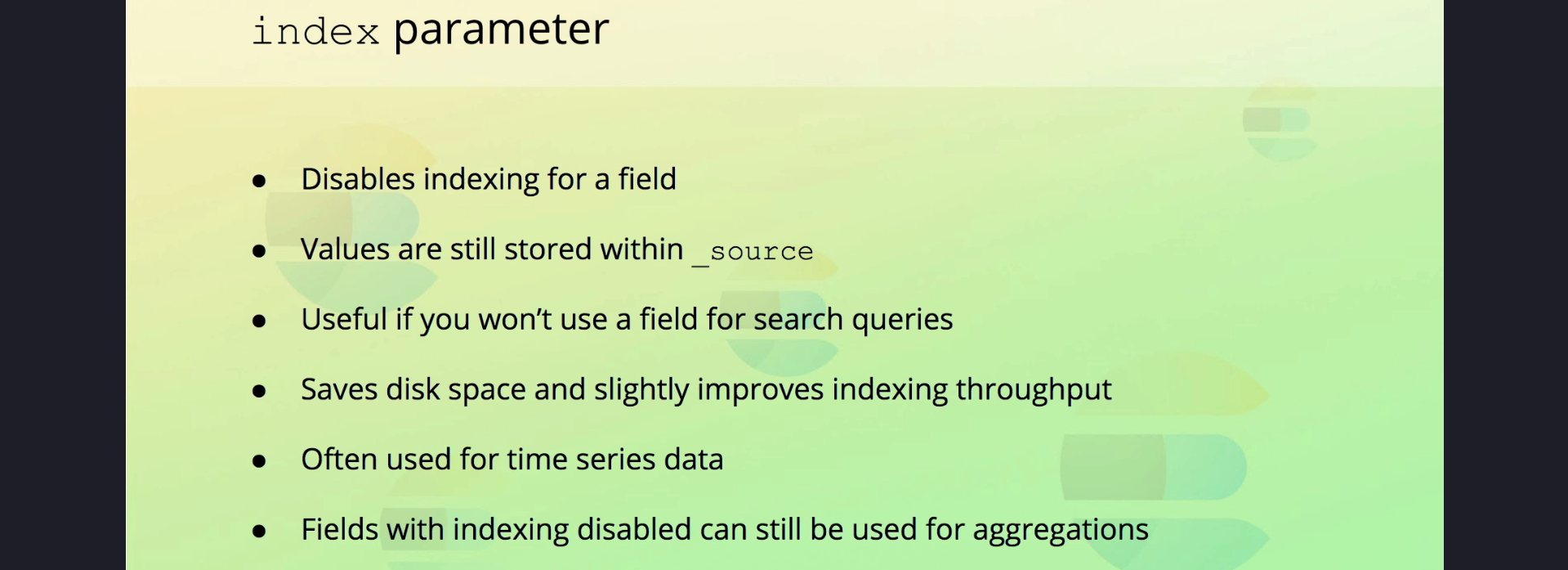
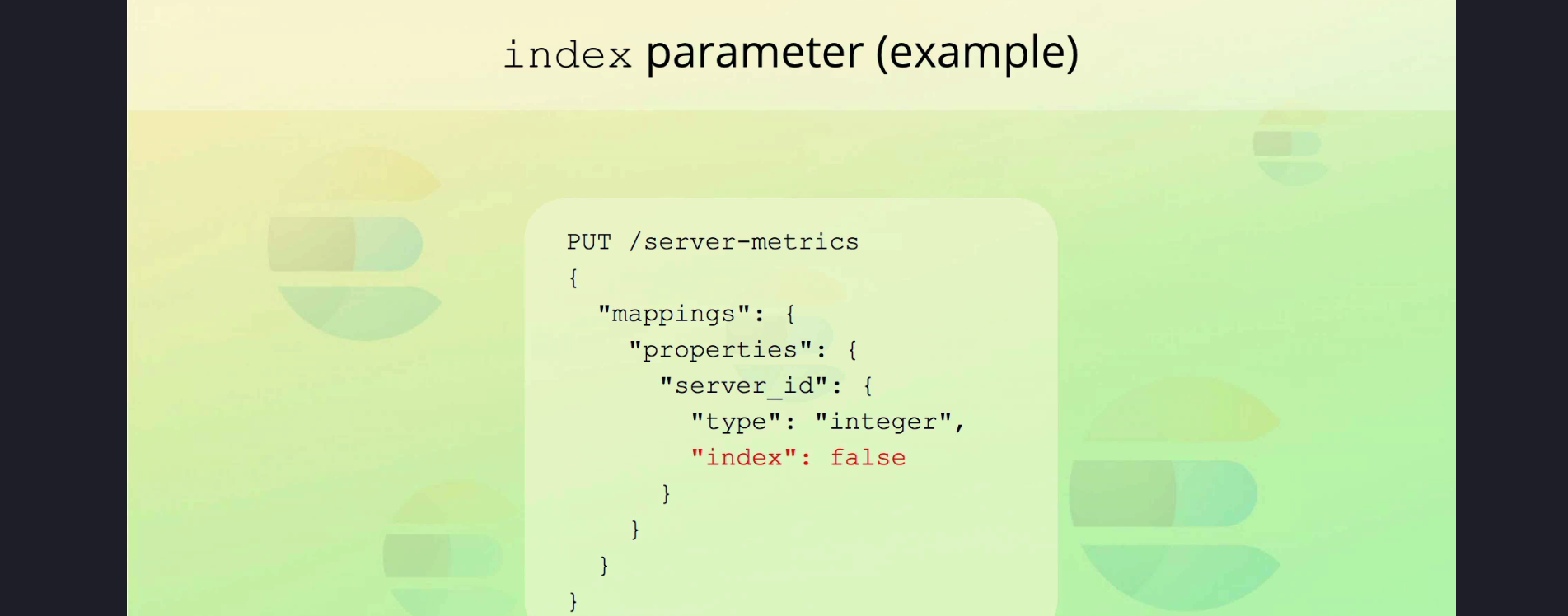
This requires us to dive a little deeper into Apache Lucene.  
We already learnt what “Inverted” index is and how it is used efficiently to look for a doc having a term. This is not the only data structure used as there is not a single data structure that efficiently serves all the purposes.

The 'doc\_values' parameter controls whether Elasticsearch creates the doc values data structure for a field. Doc values provide efficient access patterns for sorting, aggregations, and scripting but increase disk space usage. Disabling doc values can save disk space for fields not used in these scenarios. 

## 6. Norms Parameter

The 'norms' parameter enables or disables the storage of normalization factors for a field. Norms are used for relevance scoring in search queries. Disabling norms can save disk space for fields that are not used for relevance scoring, such as those used for filtering or aggregations.   
  

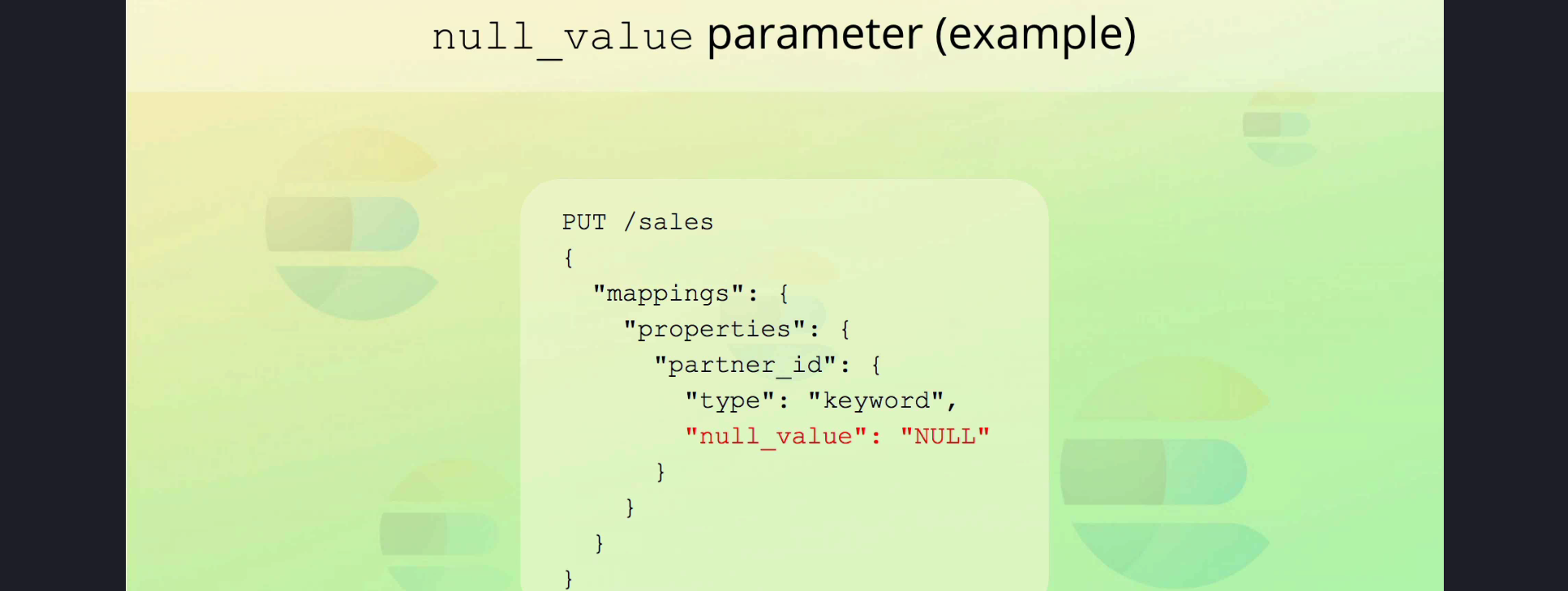
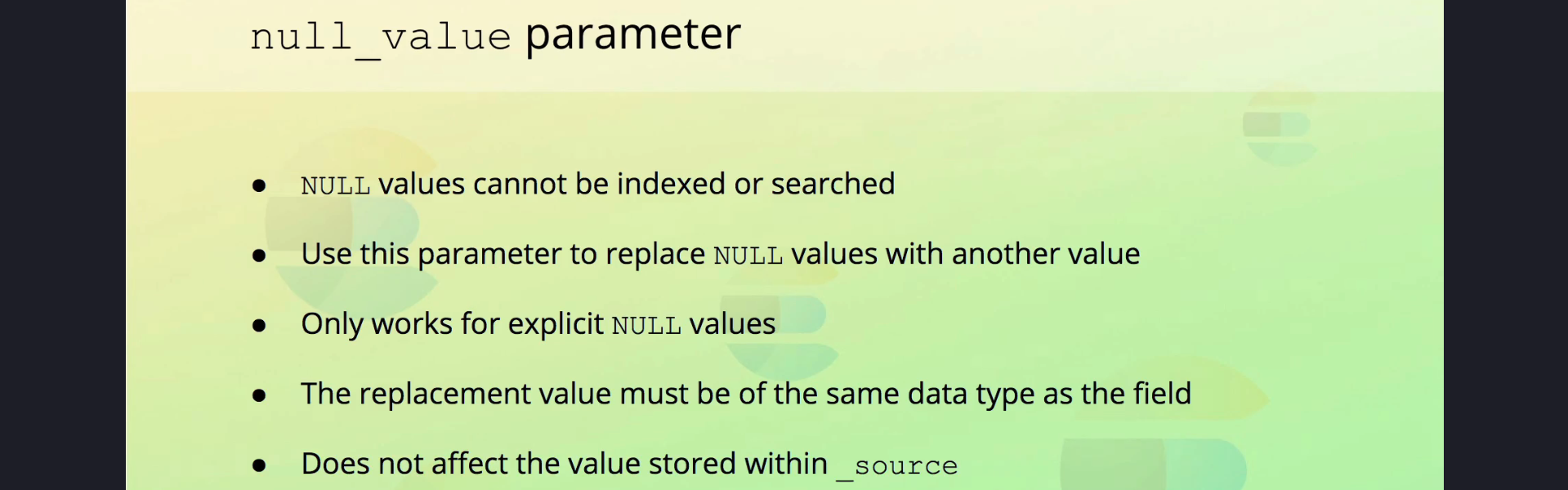

  


## 7. Index Parameter

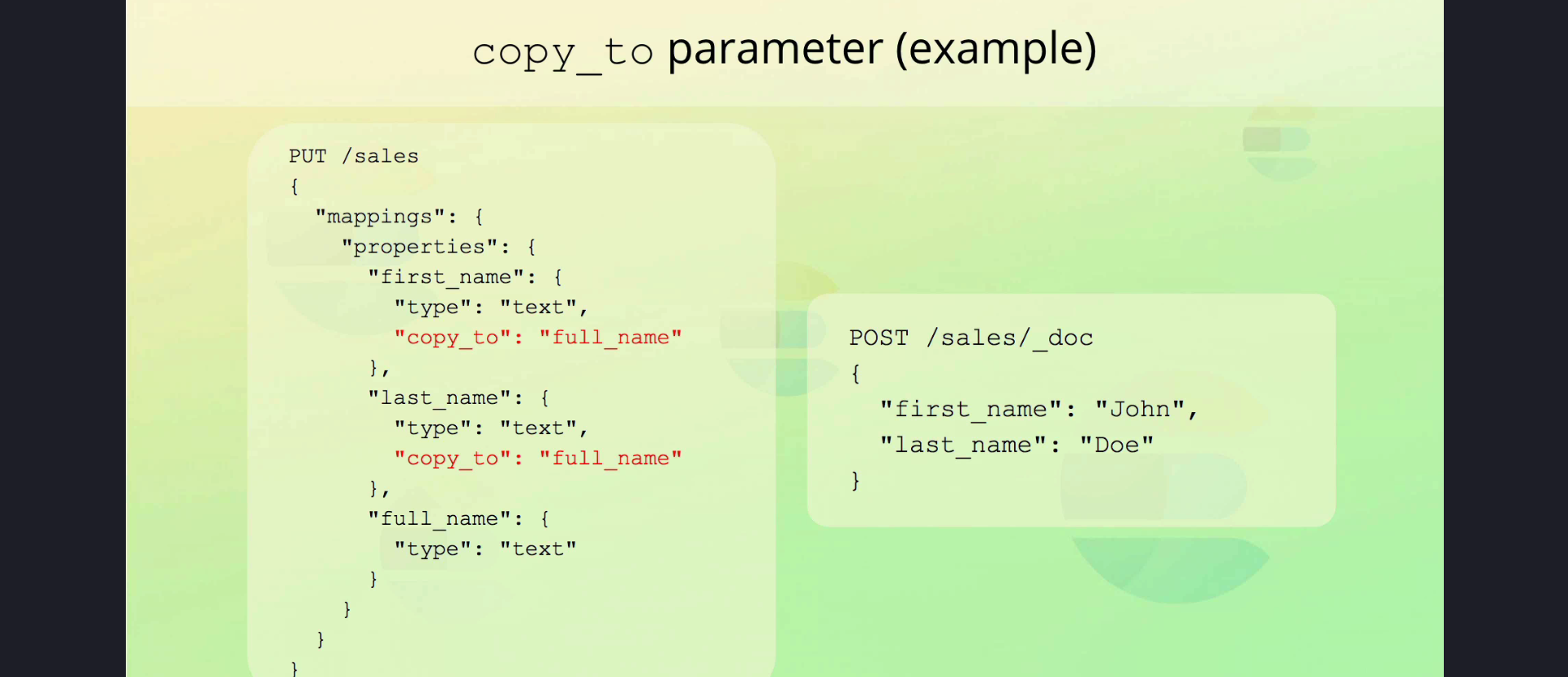
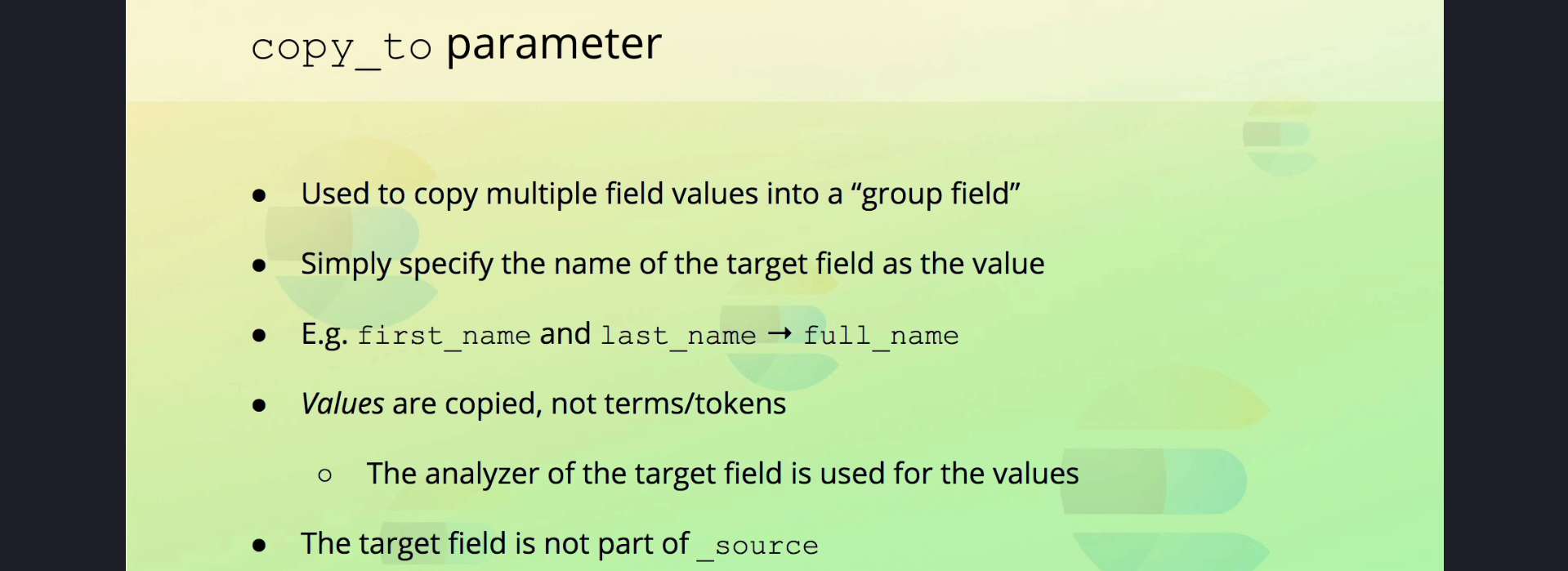
The 'index' parameter specifies whether a field should be indexed. Non-indexed fields cannot be used in search queries but remain part of the '\_source' object. This parameter is useful for fields that are stored for reference but not searched.

## 8. Null Value Parameter

The 'null\_value' parameter allows replacing NULL values with a specified value, making them searchable. The replacement value must match the field's data type. Note that this parameter applies only to explicit NULL values, not empty arrays.



## 9. Copy To Parameter

The 'copy\_to' parameter allows copying field values into another field, such as combining 'first\_name' and 'last\_name' into a 'full\_name' field. The copied values are analyzed using the target field's analyzer but do not appear in the '\_source' object unless explicitly included. s

## 10. Summary

Elasticsearch provides several parameters to customize field behavior, including those for date formats, type coercion, and relevance scoring. Understanding these parameters helps optimize data storage and query performance. While this document covered key parameters, additional advanced parameters can be explored in the Elasticsearch documentation.