# Field Aliases in Elasticsearch

## Overview

In the previous lecture, we discussed how to rename a field during reindexing, such as renaming the `content` field to `comment`. However, reindexing millions of documents just for a field rename is inefficient. This lecture introduces field aliases, which allow renaming fields without reindexing.

## What are Field Aliases?

A field alias allows you to use a different name for a field in your queries without altering the original field's mapping. This is achieved by pointing an alias to the actual field.

Key Benefits:

* • No need to reindex documents for renaming fields.
* • Allows flexible queries using the alias or the original field name.

## Adding a Field Alias

To add an alias:  
1. Define a new field mapping with the `alias` data type.  
2. Specify the target field name using the `path` parameter.

Example Query to Add Alias:

PUT /reviews/\_mapping  
{  
 "properties": {  
 "comment": {  
 "type": "alias",  
 "path": "content"  
 }  
 }  
}

## Using Field Aliases in Queries

After adding the alias, you can query either the original field or the alias.

Query 1: Search Using the Original Field:

GET /reviews/\_search  
{  
 "query": {  
 "match": {  
 "content": "great product"  
 }  
 }  
}

Query 2: Search Using the Alias:

GET /reviews/\_search  
{  
 "query": {  
 "match": {  
 "comment": "great product"  
 }  
 }  
}

Observation: Both queries yield the same results.

## How Field Aliases Work

Field aliases are a query-time construct:  
- When a query or indexing operation is executed, the alias (`comment`) is translated to the actual field (`content`).  
- This translation does not affect how values are indexed, making aliases lightweight and efficient.

Diagram Explanation:  
• Index Requests: The alias is translated into the actual field name before data is stored.  
• Search Queries: The alias is translated into the original field name before the query is processed.

## Updating a Field Alias

Field aliases are one of the few mapping constructs that can be updated. You can modify the alias's target field by performing a mapping update:

Example Query: Update Alias Target:

PUT /reviews/\_mapping  
{  
 "properties": {  
 "comment": {  
 "type": "alias",  
 "path": "new\_field"  
 }  
 }  
}

## Limitations of Field Aliases

While there are a few limitations in their usage, they are sufficient for most common scenarios. Aliases do not modify existing mappings but add flexibility for query processing.

## Beyond Field Aliases: Index Aliases

In addition to field aliases, index aliases exist at the cluster level. These:  
• Allow redirecting queries from one index to another.  
• Are especially useful for managing high volumes of data.

While not covered in detail here, index aliases are an essential concept related to data handling at scale.

## Key Takeaways

1. Field aliases allow renaming fields without reindexing.

2. They are lightweight and only affect query parsing.

3. Aliases can be updated to point to a different field.

4. For large-scale operations, index aliases can be explored.