# 

# Introduction to Mapping in Elasticsearch

## 1. Transitioning from Text Analysis to Mapping

Now that you understand the basics of text analysis and how terms are stored, it’s time to explore mapping in Elasticsearch. Mapping and analysis are closely related, as mapping defines how documents are structured and stored.

## 2. What is Mapping?

Mapping defines the structure of documents in Elasticsearch, including their fields and data types. In simpler terms, it is analogous to a table schema in a relational database.

For example, a simple table in a relational database can be mapped to Elasticsearch as follows:

Relational Database Schema:  
- ID (integer)  
- Name (string)  
- Price (decimal)

Elasticsearch Mapping:  
- id: long  
- name: text  
- price: double



While this comparison helps visualize the concept, mapping in Elasticsearch offers much more flexibility and functionality.

## 3. Key Features of Mapping

1. Defines the fields of a document and their data types.

2. Determines how documents are indexed and stored.

3. Offers advanced features beyond field definitions, making mapping a powerful tool in Elasticsearch.

## 4. Approaches to Mapping

Two ways of mappings:

1. Explicit Mapping.
2. Dynamic Mapping.

### 4.1 Explicit Mapping

1. In explicit mapping, fields and their data types are defined manually, typically during index creation.

2. Example: Defining a mapping explicitly for an index:  
PUT /products  
{  
 "mappings": {  
 "properties": {  
 "id": { "type": "long" },  
 "name": { "type": "text" },  
 "price": { "type": "double" }  
 }  
 }  
}

### 4.2 Dynamic Mapping

1. Dynamic mapping is a feature that allows Elasticsearch to automatically create field mappings when it encounters a new field.

2. The data type is determined based on the value supplied for the field. For example:  
- A string value maps to the 'text' data type.  
- A numeric value maps to a numeric data type.

3. Dynamic mapping makes Elasticsearch easier to use by eliminating the need to define every field manually.

## 5. Combining Explicit and Dynamic Mapping

1. Elasticsearch allows you to combine explicit and dynamic mapping, offering flexibility and control over field definitions.

2. Example: Explicitly define critical fields while allowing dynamic mapping for additional fields.

## 6. Summary

1. Mapping defines the structure and data types of documents in Elasticsearch, similar to a table schema in relational databases.

2. Elasticsearch supports two main approaches to mapping:  
- Explicit mapping: Manual field and data type definitions.  
- Dynamic mapping: Automatic field mapping based on data values.

3. By combining explicit and dynamic mapping, Elasticsearch provides a flexible and powerful mechanism for managing document structures.