Chapter 2
Document-oriented Databases: XML

Document-oriented Databases: XML

- XML (eXtensible Markup Language) is one of the data formats that can be used in a NoSQL context

- XML is a markup language designed to store and transport data.
- It is readable by both humans and machines, making it ideal for exchanging information between different applications.

Document-oriented Databases: XML

Syntaxe:

```
<?xml version="1.0" encoding="UTF-8"?>

<catalogue>
    <produit>
        <nom> Ordinateur portable </nom>
        <description>Ordinateur portable haut de gamme avec
            processeur rapide et grande capacité de stockage.
        </description>
        <prix>1200.00</prix>
        <quantiteStock>20</quantiteStock>
    </produit>

    <!-- Commentaire -->

</catalogue>
```

Document-oriented Databases: XML

- XML can be used in the NoSQL context:

- **Data storage** : NoSQL databases can use XML as a data storage format. Instead of using relational tables, data is stored as XML documents in a NoSQL database.

- **Flexible format** : suitable for representing complex and hie rarchical data.useful for applications where the structure may change over time.

Document-oriented Databases: XML

- XML can be used in the NoSQL context:

- **Semi-structured processing :**Store data without the need for a fixed schema.

- **XPath and XQuery queries:** To query stored XML data.Flexibility in querying semi-structured data.

Document-oriented Databases: XML

- XML Validation :
- XML documents can be validated against a schema (DTD, XML Schema)
- Ensuring their structural conformity.

```
<?xml version="1.0"?>
 <!DOCTYPE bibliotheque [
<!ELEMENT bibliotheque (livre+)>
<!ELEMENT livre (titre, auteur, ref)>
 <!ELEMENT titre (#PCDATA)>
<!ELEMENT auteur (nom, prenom)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
<!ELEMENT ref (#PCDATA)> ]>
<bibliotheque>
 ...
 </bibliotheque>
```

Document-oriented Databases: XML

- XML Transformation:

XSLT (eXtensible Stylesheet Language Transformations)to transform and present XML data in a different way.To other formats: html, PDF, Doc, ….

Document-oriented Databases: XML

- **Exercise 1**

We want to create a product catalog for an E-commerce site. Each product must have a name, a description, a price and a quantity in stock. Create an XML document representing three products

Create the DTD Document that validates the XML Document

- **Exercise 2**

Create an XML file that models the management of books in a library. The file contains information about books, authors, and user loans.

(library contains: books, authors, loans)

(books contains: title, isbn, publicationdate, authorid)

(authors contains: id, name, nationality)

(loans contains: isbn, username, loandate, returndate)

Document-oriented Databases: XML

- **Exercise 3**

Create an XML file that models the trips offered by a travel agency

The file must have a root element called <travel agency>.

The main sub-elements must include :

<destinations> : <id>, <name>, ,<country>, <description>

<trips> : <id> <idDestination>, <dateDeparture>, <dateReturn> , <price>

<clients> : <id> , <name>, <email> ,<tripsRegistered>

Document-oriented Databases: XML MarkLogic

- **MarkLogic DataBase**

- Is an example of a NoSQL database specifically designed to handle XML data.

- It offers XML-based search and content management capabilities.

- MarkLogic is a multi-model NoSQL DB:XML documents, JSON documents, RDF, Plain Text,….and other semi-structured data formats.

- It is designed to manage, store and search complex and heterogeneous data

Document-oriented Databases: XML MarkLogic

- MarkLogic has powerful text search
- and semantic search capabilities.
- It automatically indexes data for easy searching and querying.
- It provides semantic capabilities to draw conclusions from stored data
- Manage complex data with hierarchical structures
- Highly scalable, with load balancing and parallelism mechanisms

Document-oriented Databases: XML Elasticsearch

- Elasticsearch is a NoSQL database,
- belongs to the category of distributed search engines
- Initially designed for text search
- Elasticsearch has evolved into a versatile tool
- Used in various fields, including semi-structured and unstructured data management.

Document-oriented Databases: XML Elasticsearch

- Some key points about Elasticsearch as a NoSQL DB:

1. Elasticsearch stores data as JSON documents.
2. It uses the Apache Lucene search engine to perform text searches
3. Elasticsearch is designed to be distributed, ensuring horizontal scalability to handle large amounts of data.

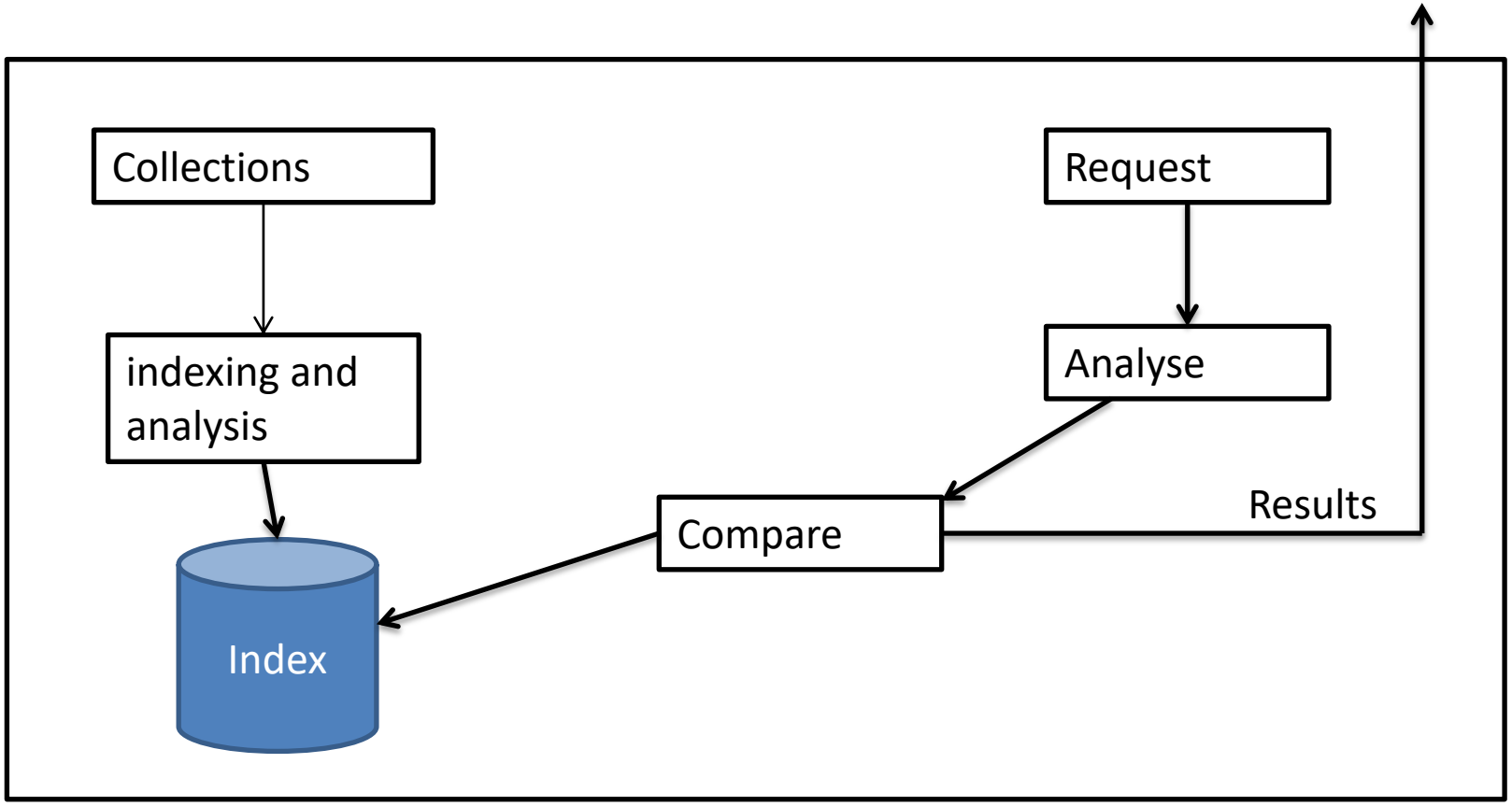
Document-oriented Databases: XML Elasticsearch

- Some key points about Elasticsearch as a NoSQL DB:

4. Elasticsearch supports aggregations, allowing to analyze data to obtain statistical information,
5. Elasticsearch is used for full-text search, log analysis, real-time monitoring, geospatial data search,

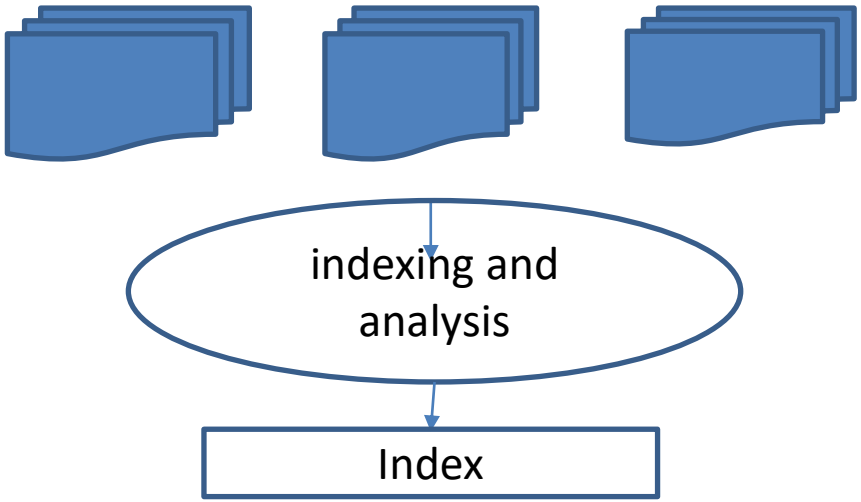
Document-oriented Databases: XML Elasticsearch

- Elasticsearch system :



Document-oriented Databases: XML Elasticsearch

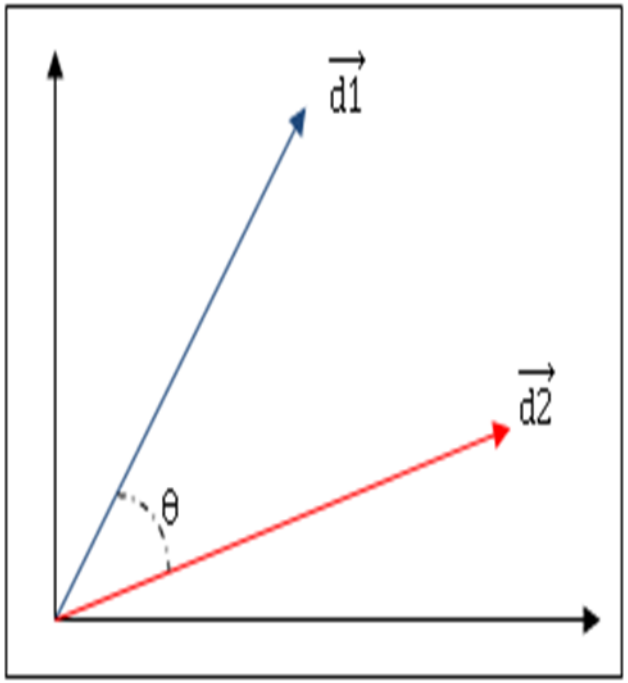
indexing and analysis of data collection



The	Doc1, Doc2, Doc3, Doc4, Doc5,…..
data	Doc1, Doc2, Doc3, Doc4, Doc5,…..
Retrieval	Doc2, Doc5
Information	Doc2, Doc5
Base	Doc3, Doc5

Document-oriented Databases: XML Elasticsearch

- Comparison between Documents
- $\text{Cos}(\theta) = \frac{\overrightarrow{d_1} \cdot \overrightarrow{d_2}}{|\overrightarrow{d_1}|\,|\overrightarrow{d_2}|}$



Documents and queries are represented as vectors

D1={w11, w21,....., w N, 1 },

q= { w1,q, w2,q,....., w N, q }

$\overrightarrow{d_1} \cdot \vec{q} = \sum wd1 * wq$

Norme $|d_1| = \sqrt{w1^2 + w2^2 + w3^2}$ ….

Document-oriented Databases: XML Elasticsearch

- Vector Vocabulary:
- [Quality, Battery Life, Value for Money, Ease of Use, Camera, Sound, Design, Display, Speed, Reliability,……]
- customer 1= {0.8, 0.6, 0.9, 0.5, 0.8};

- Product 1: {0.6, 0.8, 0.7, 0.9, 0.5}, ex: Galaxy S21
- Product 2:{0.2, 0.5, 0.2, 0.7, 0.6}, ex: iPhone 12
- Product 3:{0.7, 0.9, 0.6, 0.8, 0.7}, ex: Xiaomi Mi 11

Document-oriented Databases: XML Elasticsearch

- Indexing and representation

1. Lexical analysis

2. Elimination of Stop Words (the, a, of, ……)

3. Lemmatization

“informed”, “informs”, “informe”, “information” ➔ inform

Document-oriented Databases: XML Elasticsearch

- Indexation et représentation

4. Weighting of terms

Associate each term with a weight that represents its importance in the collection of documents

Two types of weighting: local and global

- local weighting: the importance of a term in a single document➔ the TF frequency
- global weighting: the importance of a term in the entire collection of documents It allows to reduce the importance of terms that are often found in a document IDF (Inverse of Document Frequency)

Document-oriented Databases: XML Elasticsearch

Advantage of document analysis and indexing:

- → to facilitate research
- → for document classification
- → Sentiment analysis
- → document recommendation