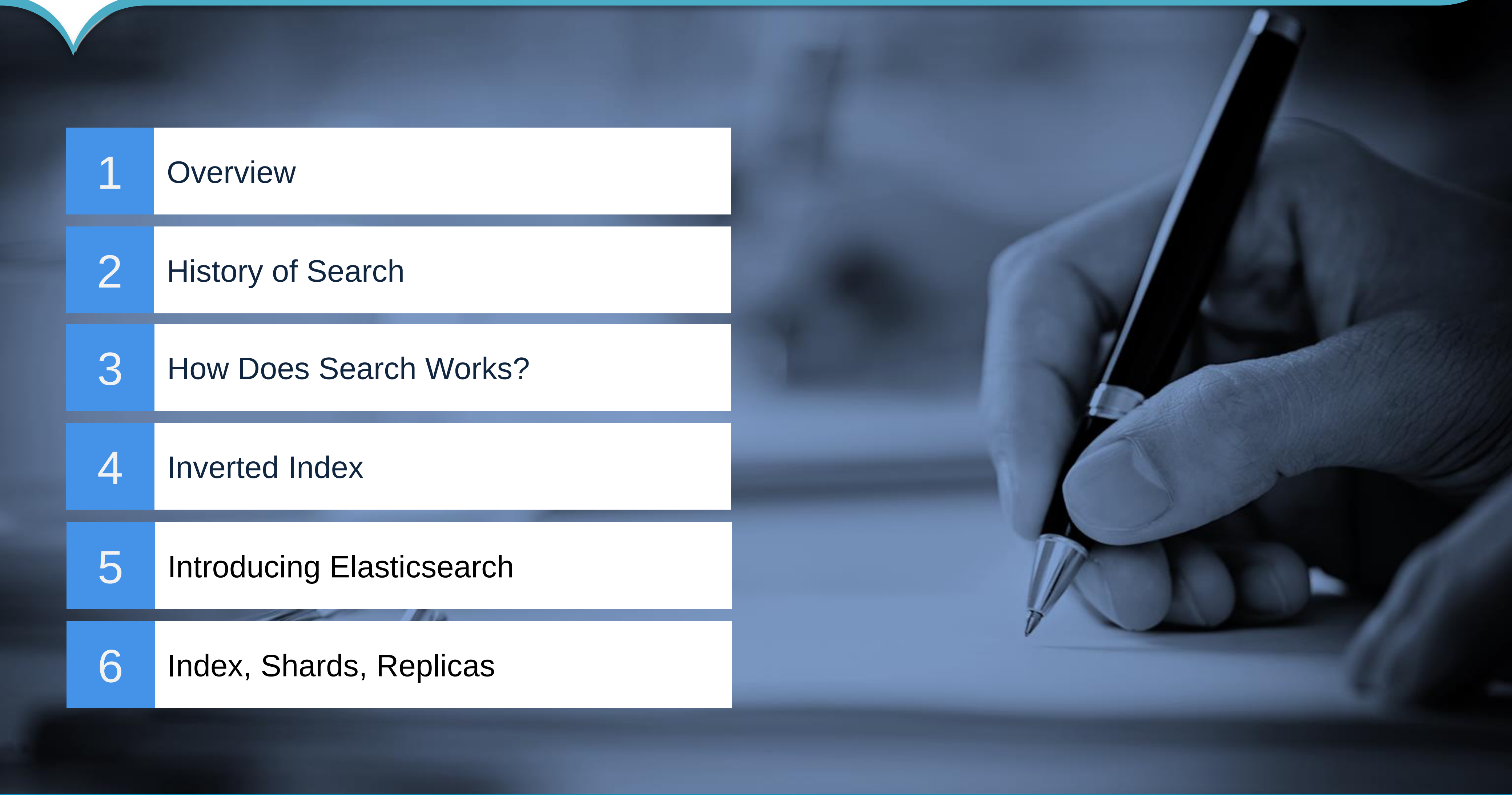
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

ElasticSearch

May 2017

People matter, results count.

Course Map - Searching and Analyzing Data with Elasticsearch



1 Overview

2 History of Search

3 How Does Search Works?

4 Inverted Index

5 Introducing Elasticsearch

6 Index, Shards, Replicas

Overview

- A little search engine history and the importance of search
- Basics steps involved in indexing and searching documents
- The inverted index, the heart of a search engine
- An introduction to Elasticsearch and its basic building blocks
- Set up and install Elasticsearch on your local machine and check cluster health

Overview

Prerequisites

- Familiarity with the command line on a Mac, Linux or Windows machine
- Familiarity with using RESTful APIs to perform actions
- A very basic understanding of distributed computing

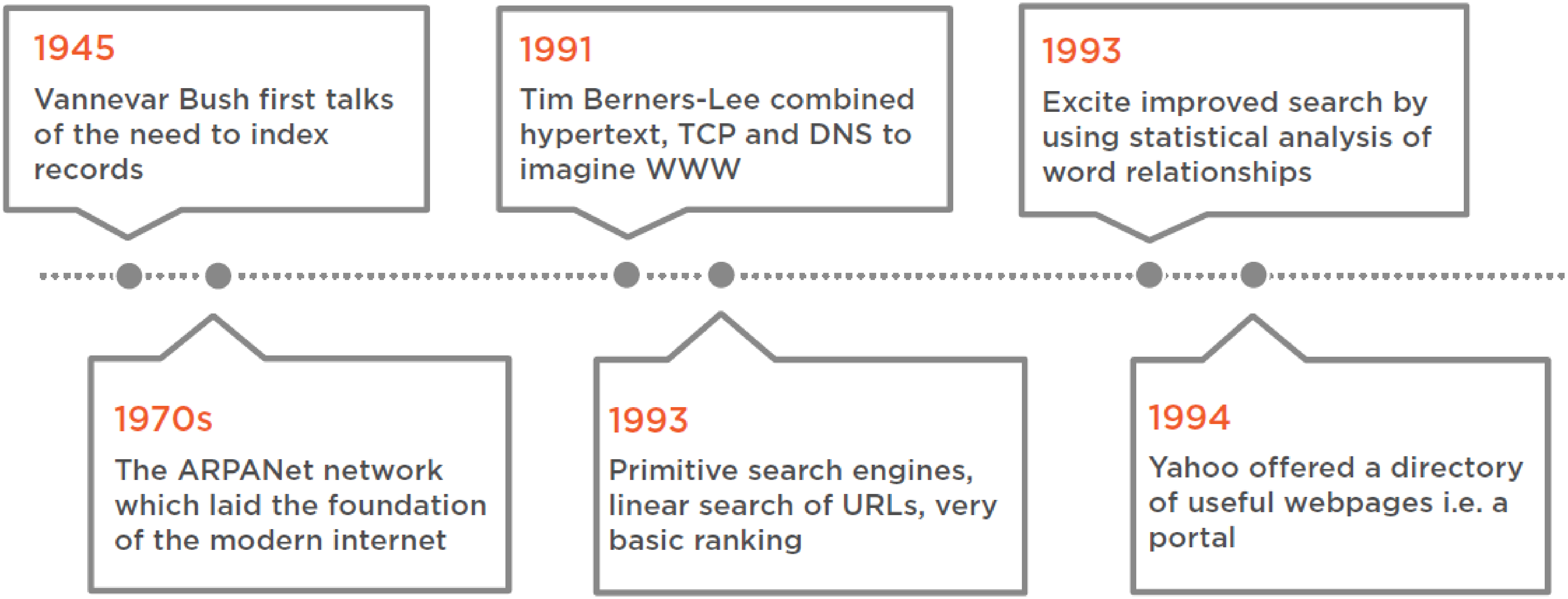
Install and Setup

- The latest version of Elasticsearch, 5.4.0 requires Java version 8
- A Mac, Linux or Windows machine on which Elasticsearch can be installed

Course Overview

- Introduction to basic concepts in Elasticsearch, download and install
- Building an index, adding documents to it both individually and in bulk
- Search queries on an index using the Query DSL
- Analysis of data on an index using aggregations

Brief History of Search



1945

Vannevar Bush first talks of the need to index records

1970s

The ARPANet network which laid the foundation of the modern internet

1991

Tim Berners-Lee combined hypertext, TCP and DNS to imagine WWW

1993

Primitive search engines, linear search of URLs, very basic ranking

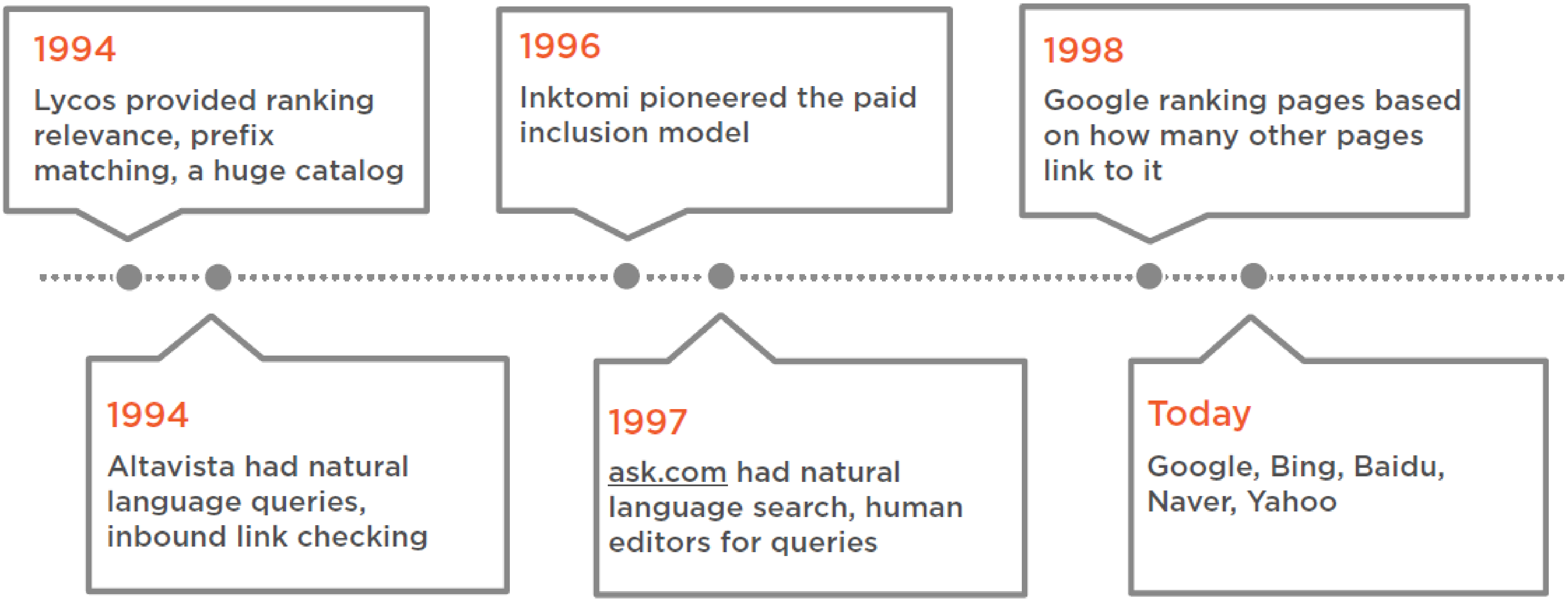
1993

Excite improved search by using statistical analysis of word relationships

1994

Yahoo offered a directory of useful webpages i.e. a portal

Brief History of Search



1994

Lycos provided ranking relevance, prefix matching, a huge catalog

1994

Altavista had natural language queries, inbound link checking

1996

Inktomi pioneered the paid inclusion model

1997

ask.com had natural language search, human editors for queries

1998

Google ranking pages based on how many other pages link to it

Today

Google, Bing, Baidu, Naver, Yahoo

How Does Search Work?

- **What Is the Objective of Search?**
  - Find the most relevant documents with your search terms

Most Relevant Document for Search Terms:

Know of the document's existence

Index the document for lookup

Know how relevant the document is

Retrieve ranked by relevance

How Does Search Work?

Most Relevant Document for Search Terms

Web crawler

Inverted index

Scoring

Search

How Does Search Work?

Search is not restricted to the Web

Sites Have Their Own Search

E-commerce

Video

E-learning

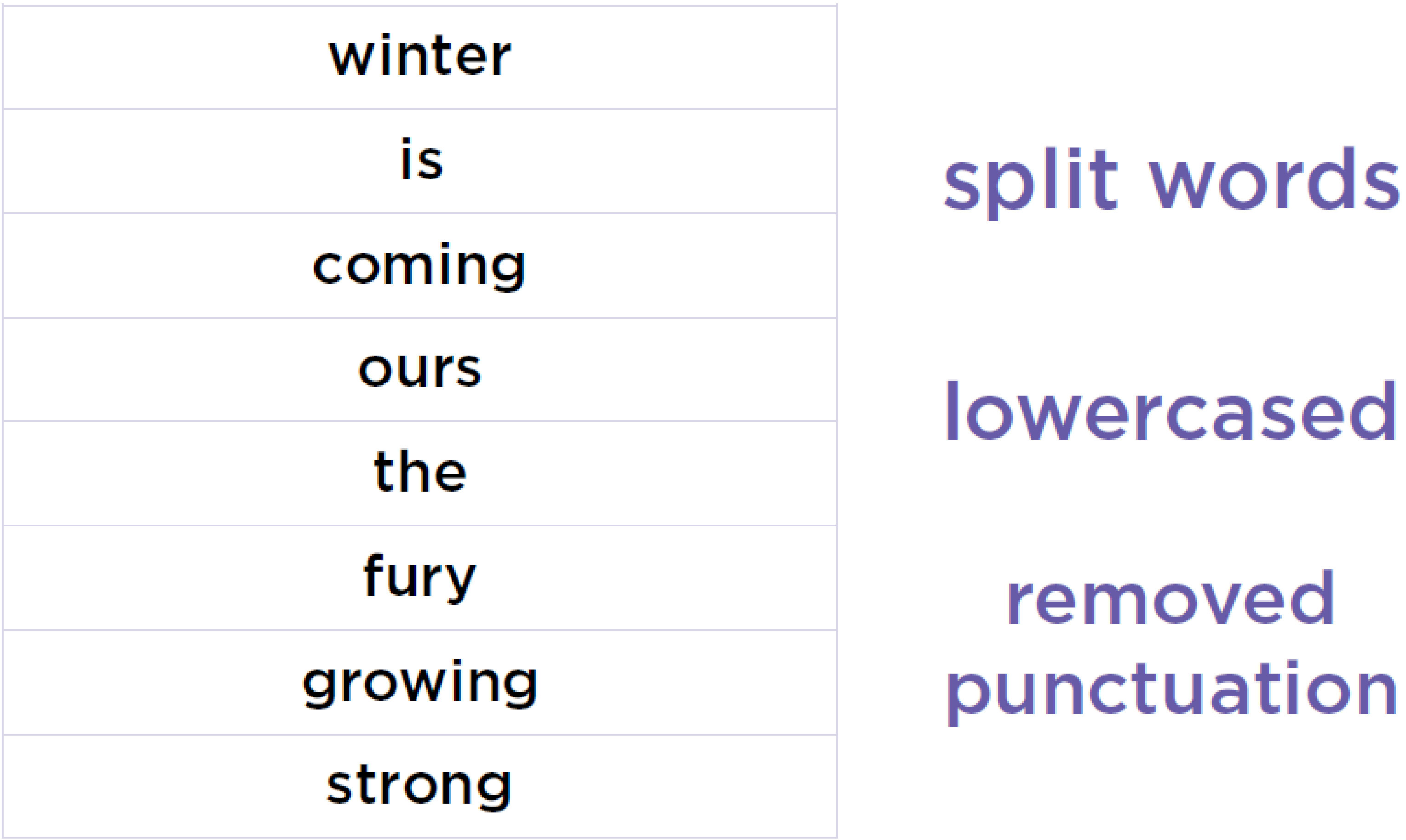
Inverted Index

Documents Have Content

House Stark	House Baratheon	House Tyrell
Winter is coming	Ours is the fury	Growing Strong

Inverted Index

Tokenize Text into Words



winter
is
coming
ours
the
fury
growing
strong

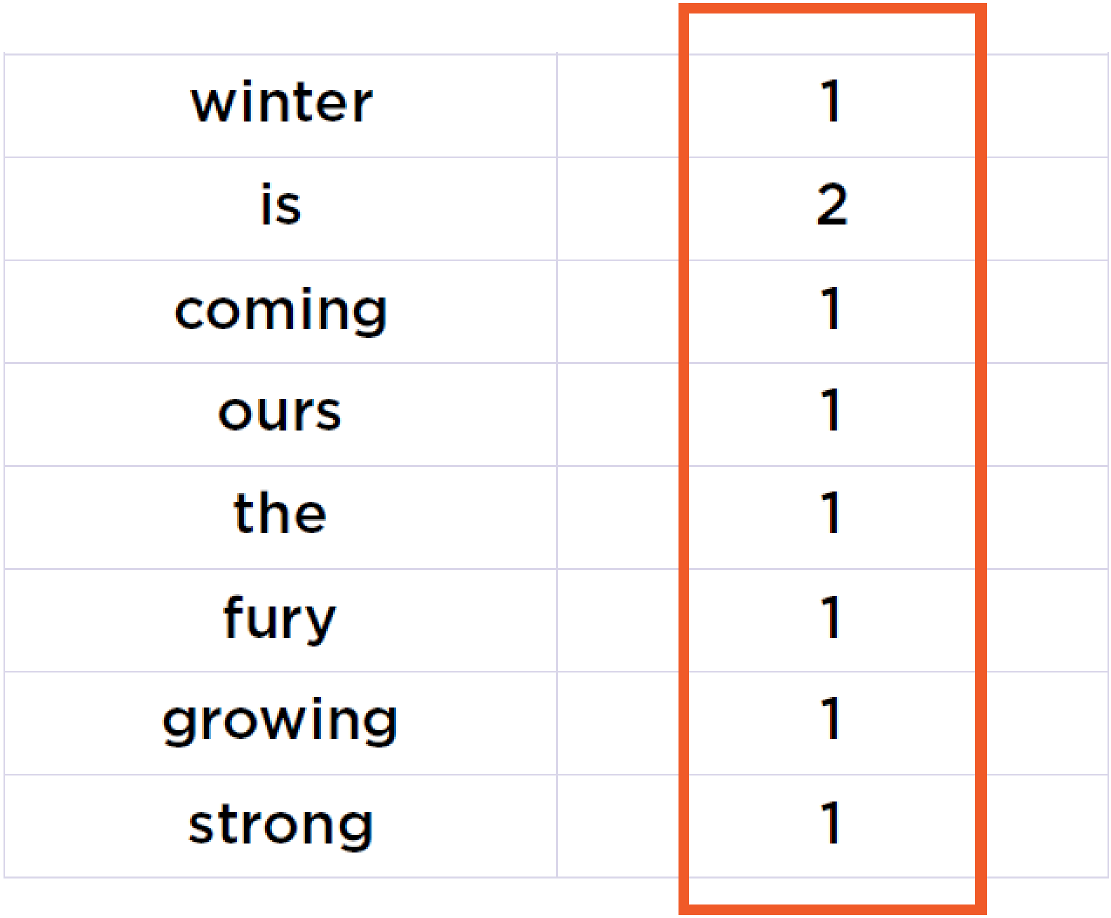
split words

lowercased

removed punctuation

Inverted Index

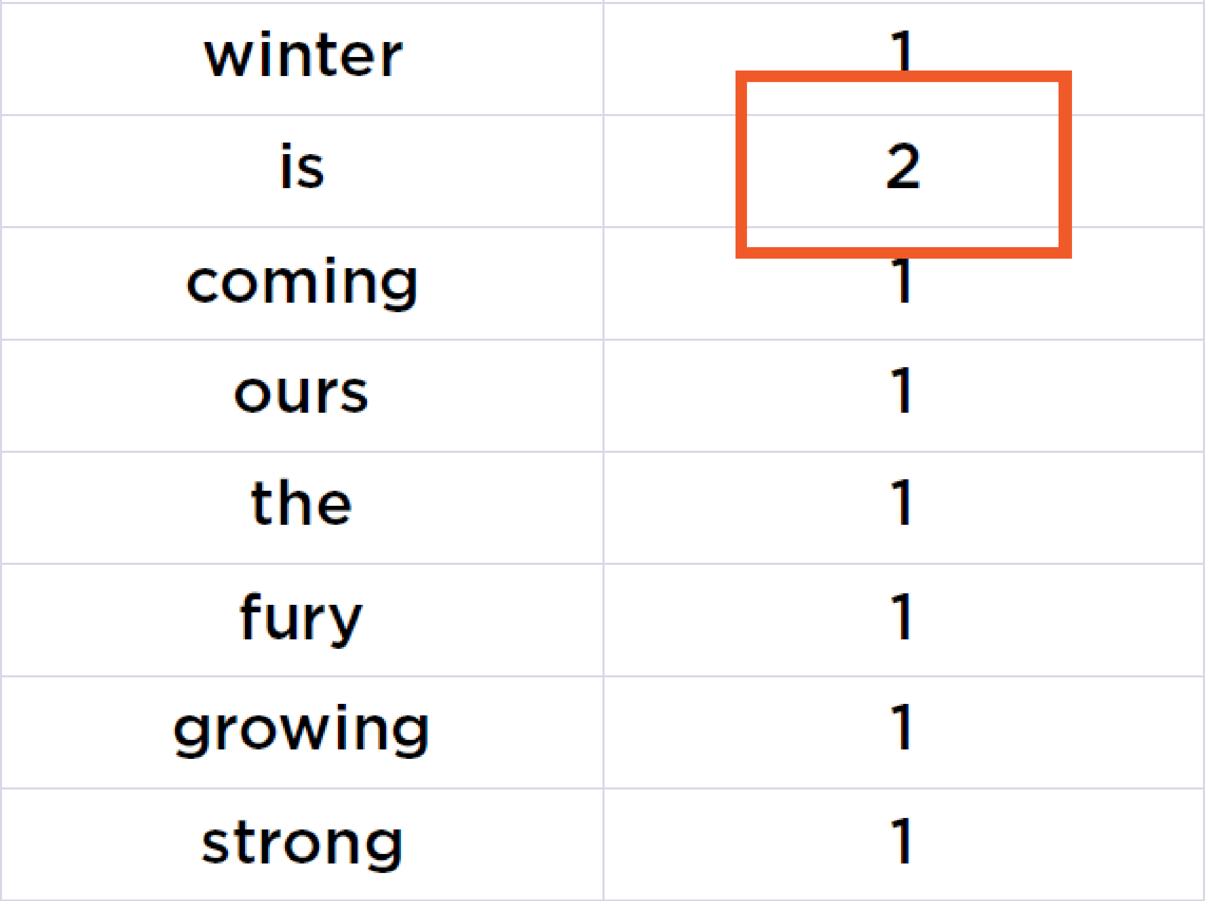
Tokenize Text into Words



winter	1
is	2
coming	1
ours	1
the	1
fury	1
growing	1
strong	1

Inverted Index

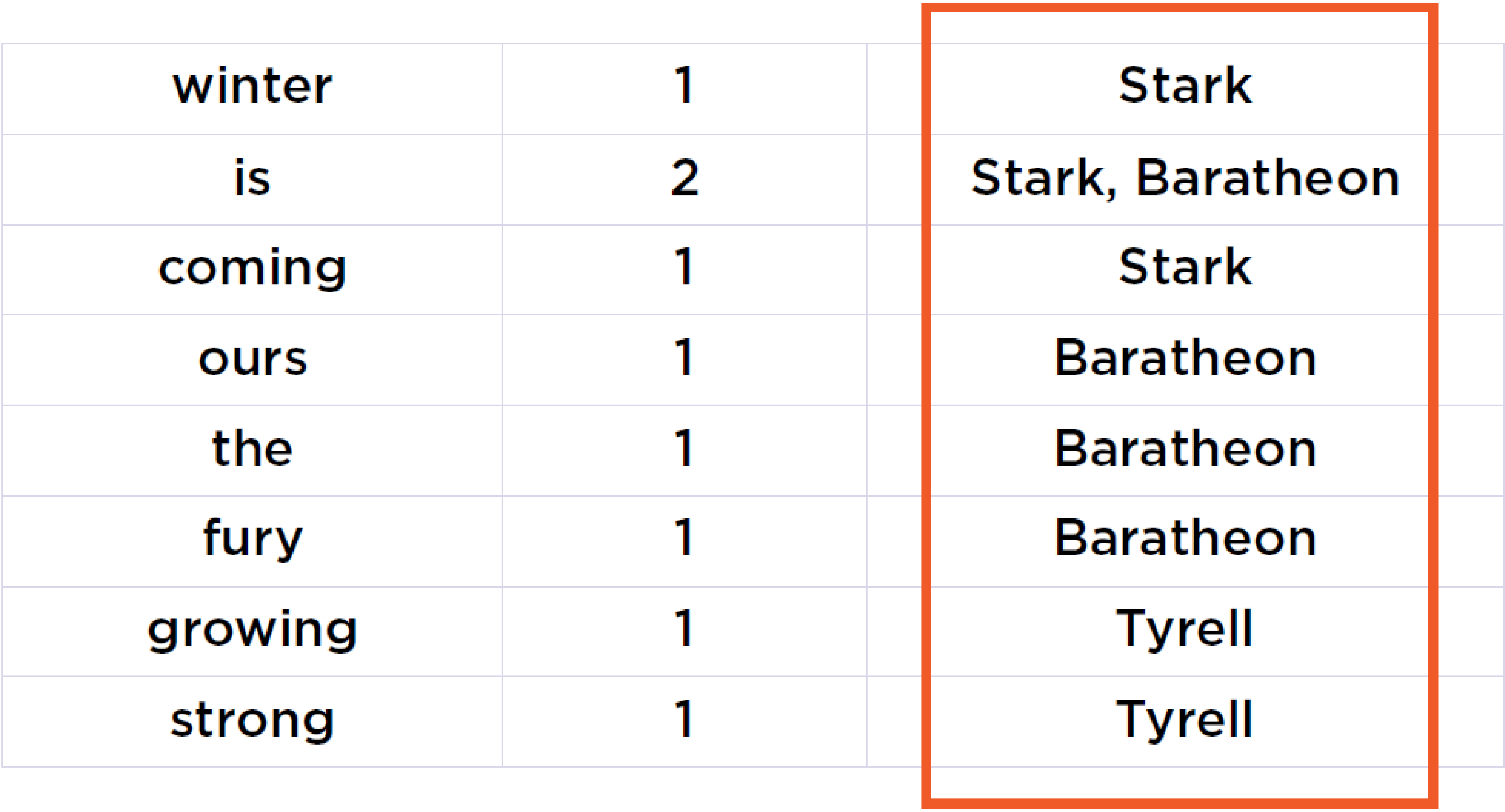
Tokenize Text into Words



winter	1
is	2
coming	1
ours	1
the	1
fury	1
growing	1
strong	1

Inverted Index

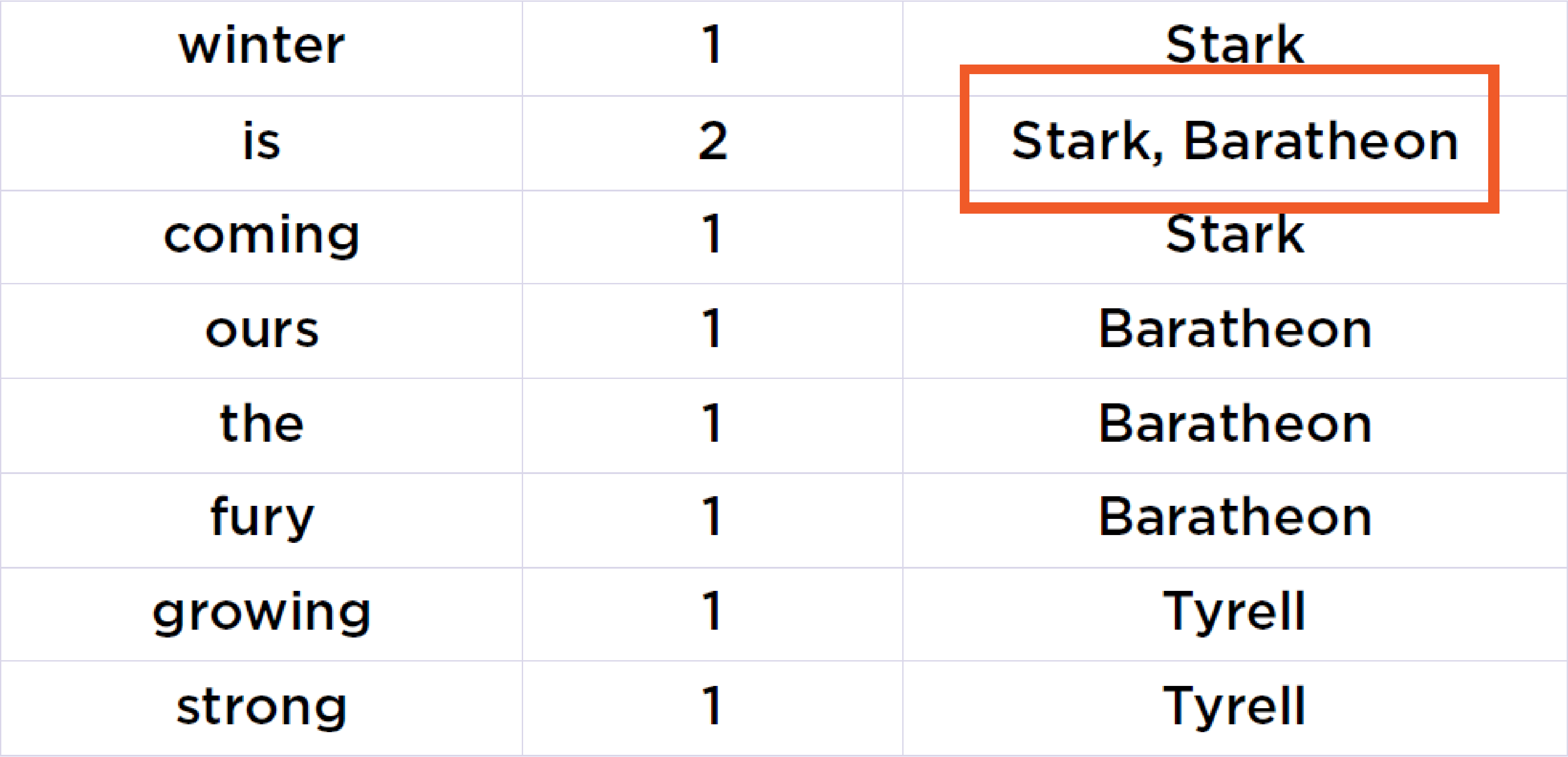
Tokenize Text into Words



winter	1	Stark
is	2	Stark, Baratheon
coming	1	Stark
ours	1	Baratheon
the	1	Baratheon
fury	1	Baratheon
growing	1	Tyrell
strong	1	Tyrell

Inverted Index

Tokenize Text into Words

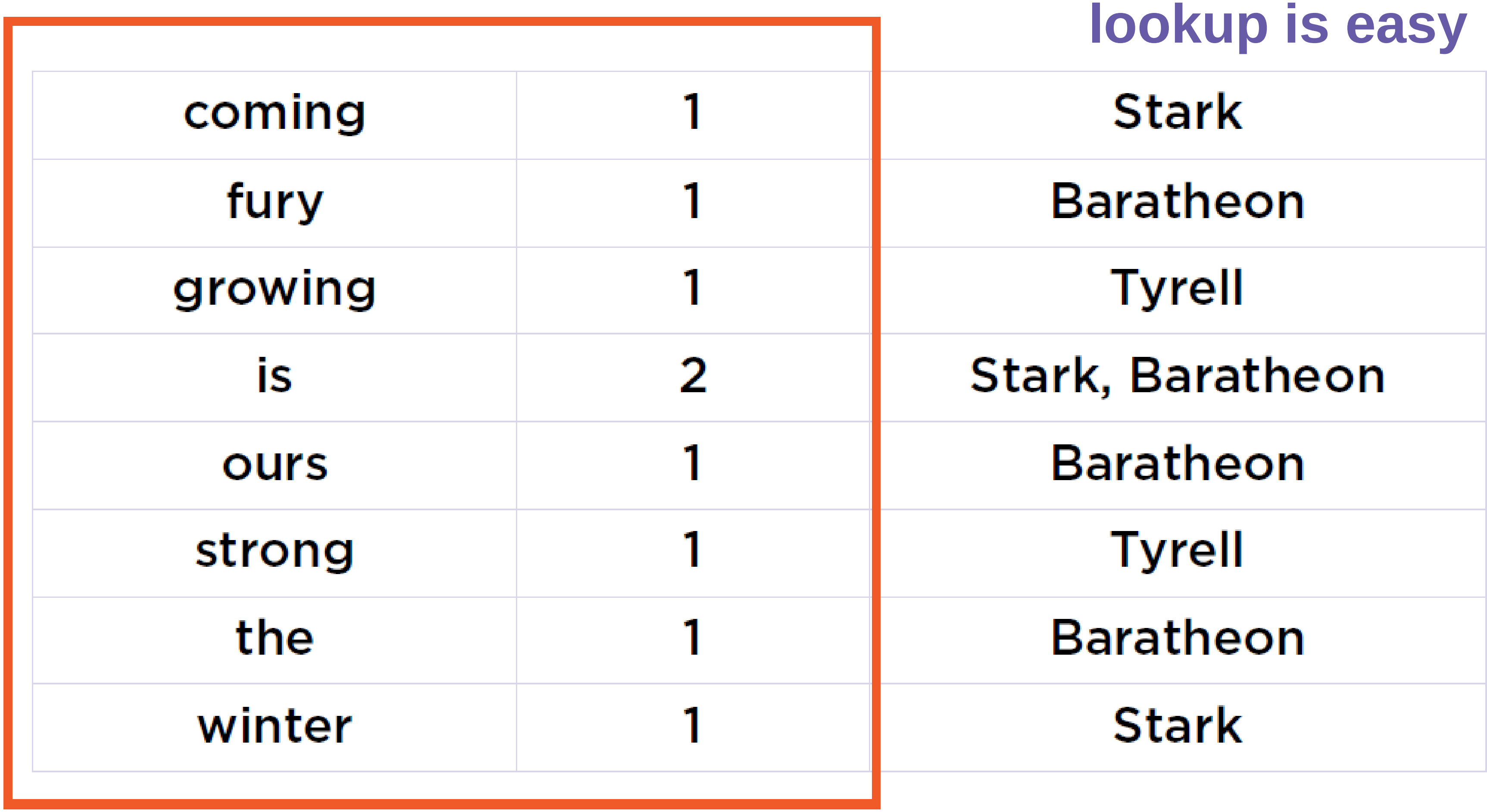


winter	1	Stark
is	2	Stark, Baratheon
coming	1	Stark
ours	1	Baratheon
the	1	Baratheon
fury	1	Baratheon
growing	1	Tyrell
strong	1	Tyrell

Inverted Index

Dictionary

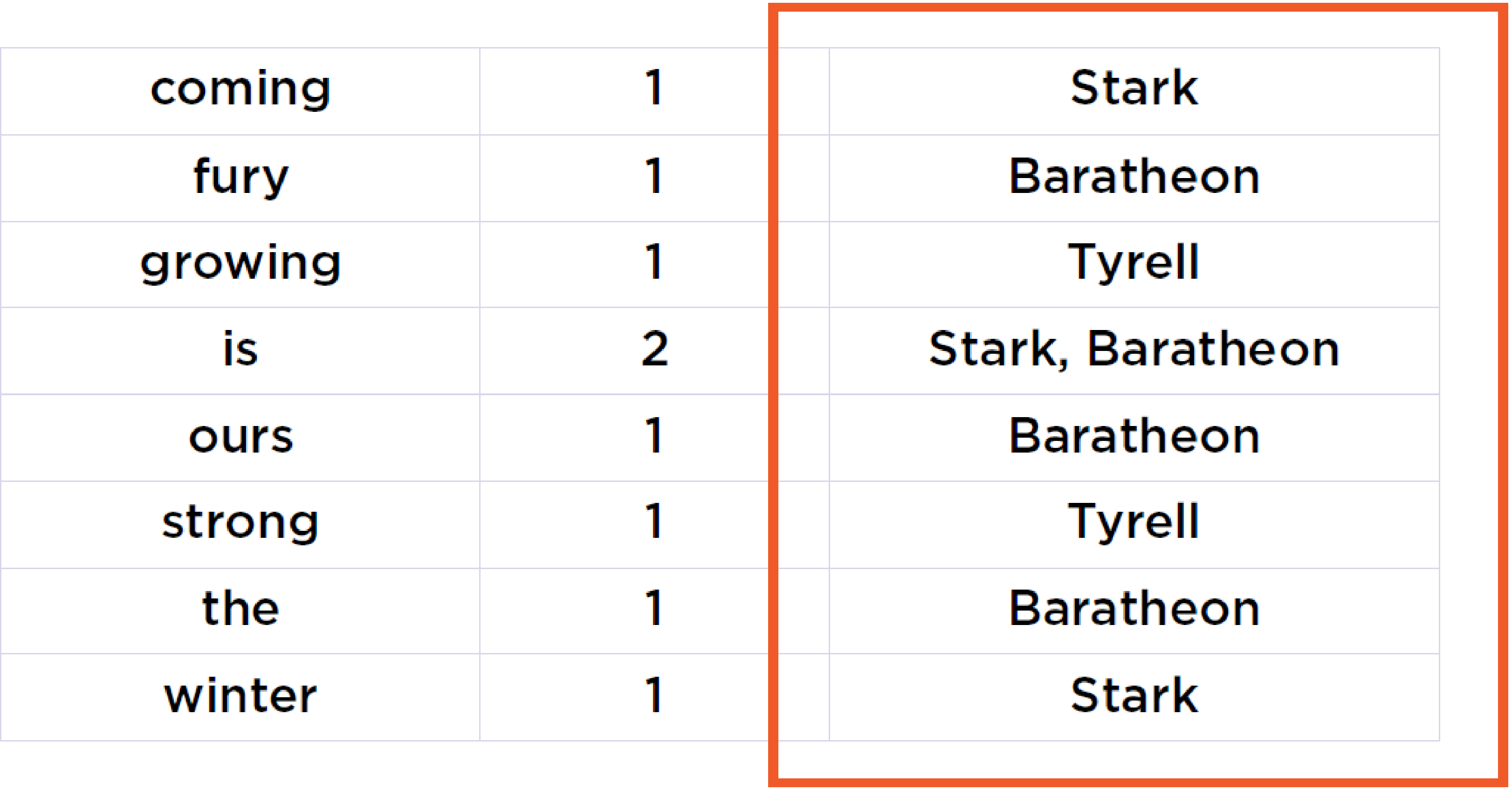
sorted so lookup is easy



coming	1	Stark
fury	1	Baratheon
growing	1	Tyrell
is	2	Stark, Baratheon
ours	1	Baratheon
strong	1	Tyrell
the	1	Baratheon
winter	1	Stark

Inverted Index

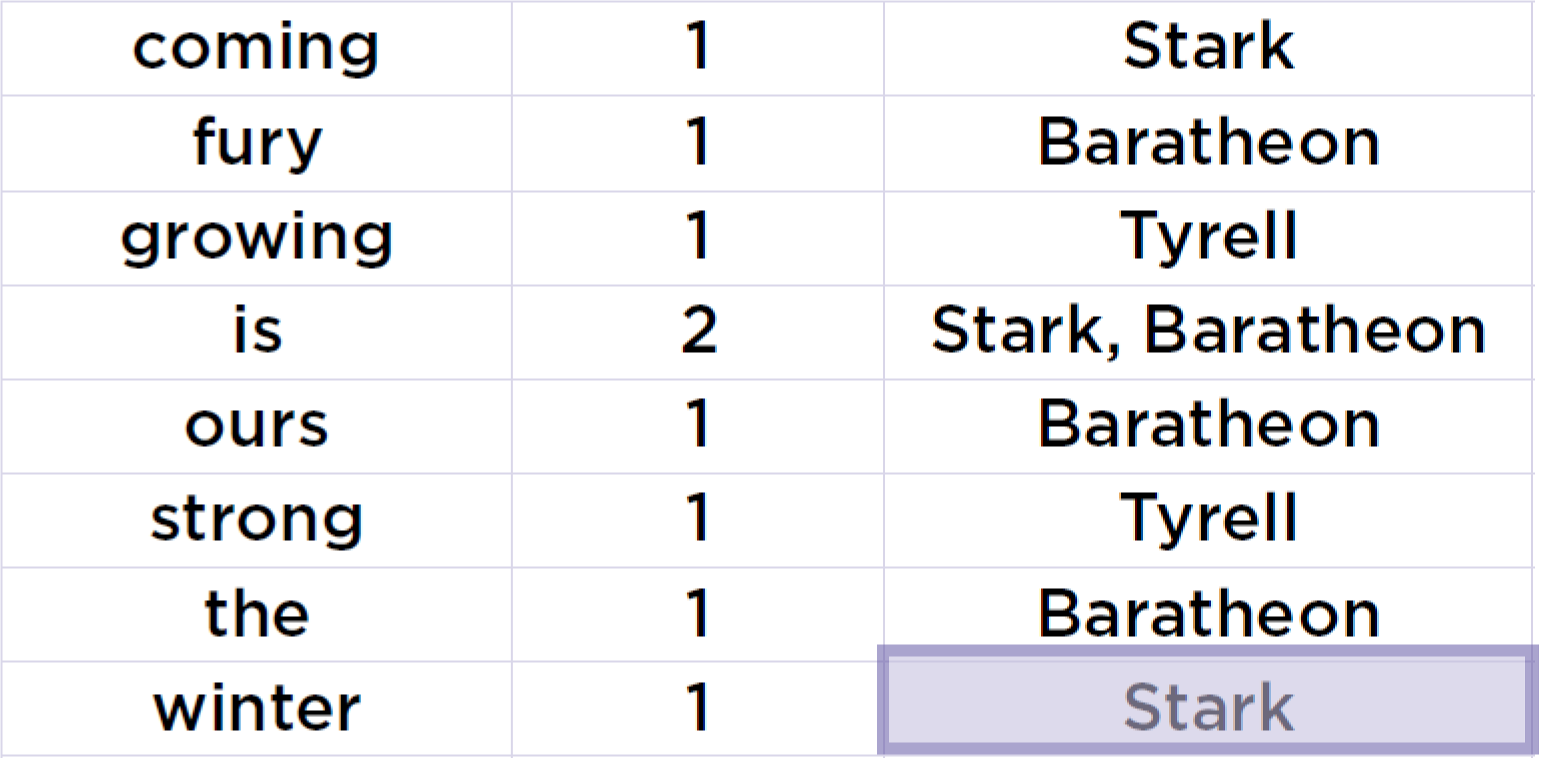
Postings



coming	1	Stark
fury	1	Baratheon
growing	1	Tyrell
is	2	Stark, Baratheon
ours	1	Baratheon
strong	1	Tyrell
the	1	Baratheon
winter	1	Stark

Inverted Index

Search

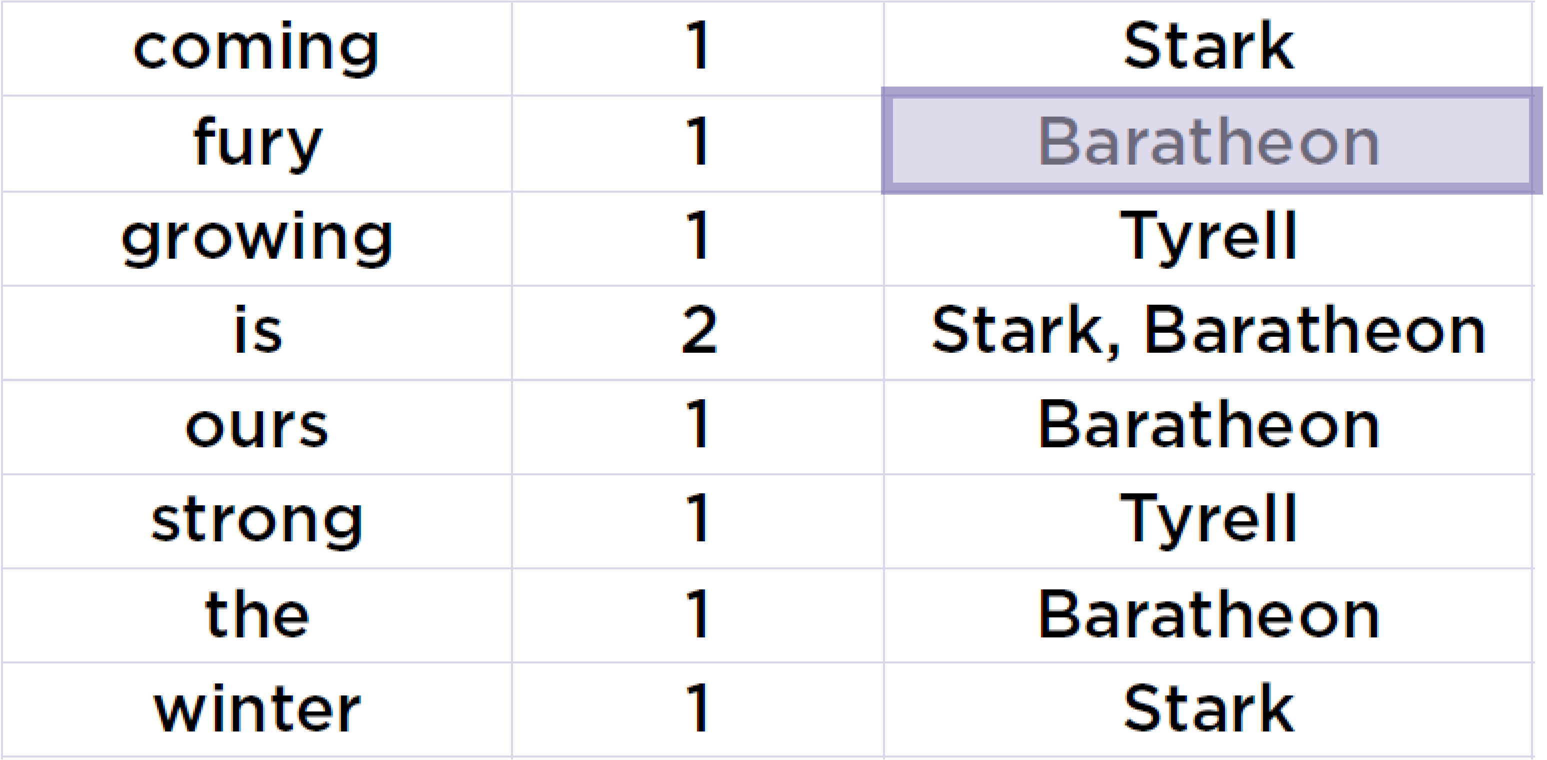


coming	1	Stark
fury	1	Baratheon
growing	1	Tyrell
is	2	Stark, Baratheon
ours	1	Baratheon
strong	1	Tyrell
the	1	Baratheon
winter	1	Stark

winter

Inverted Index

Search

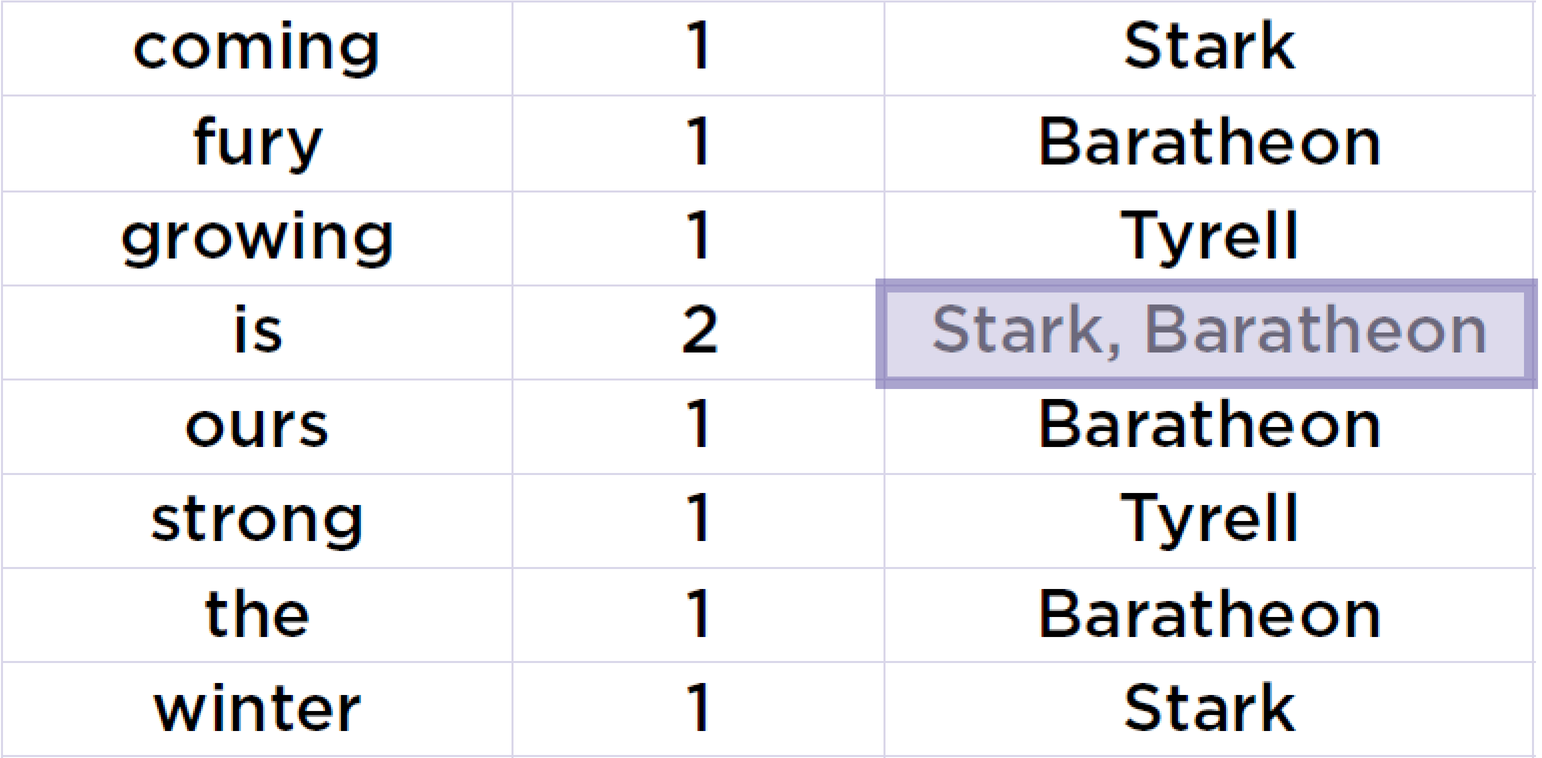


coming	1	Stark
fury	1	Baratheon
growing	1	Tyrell
is	2	Stark, Baratheon
ours	1	Baratheon
strong	1	Tyrell
the	1	Baratheon
winter	1	Stark

fury

Inverted Index

Search

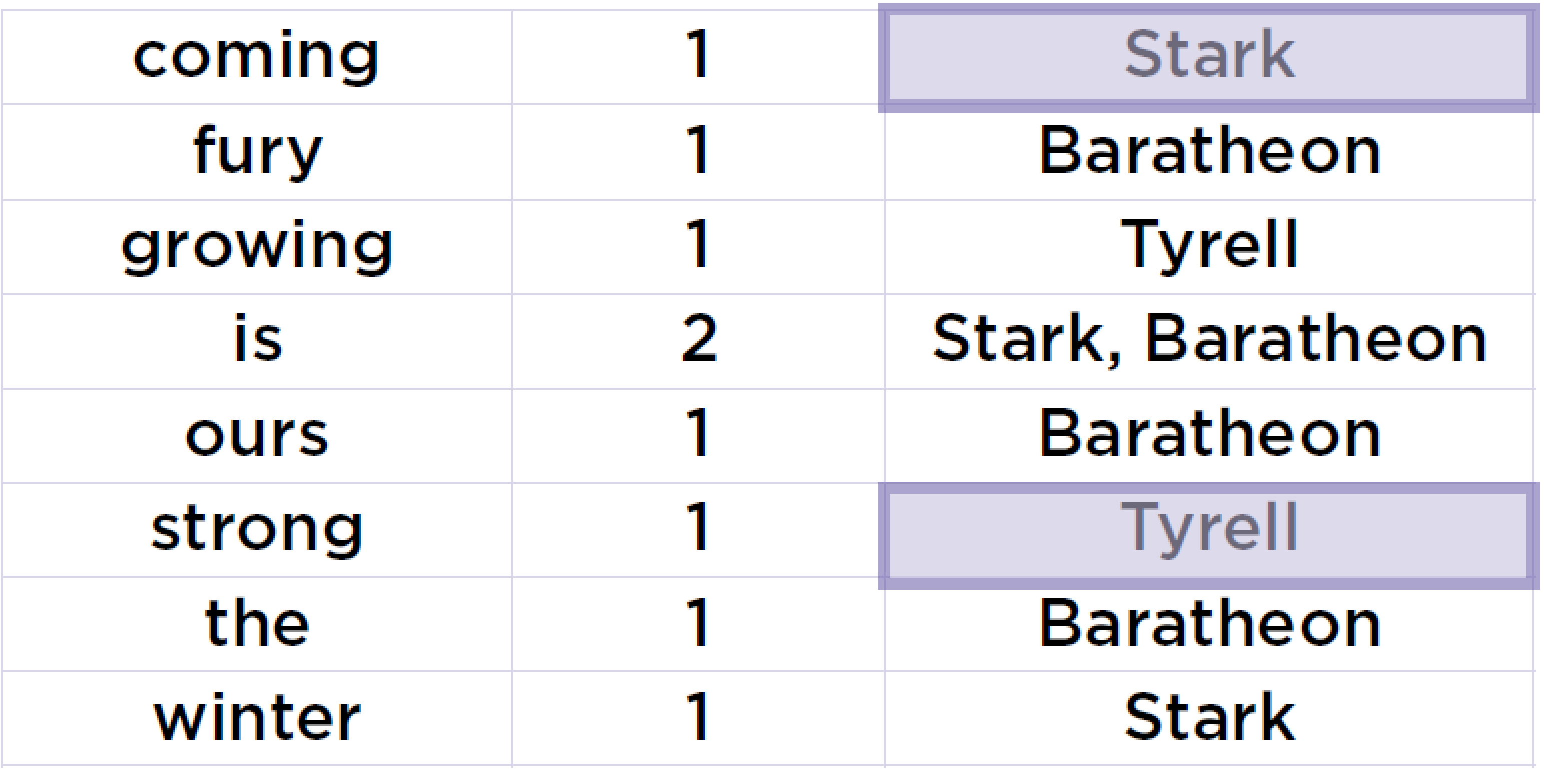


coming	1	Stark
fury	1	Baratheon
growing	1	Tyrell
is	2	Stark, Baratheon
ours	1	Baratheon
strong	1	Tyrell
the	1	Baratheon
winter	1	Stark

is

Search

Inverted Index

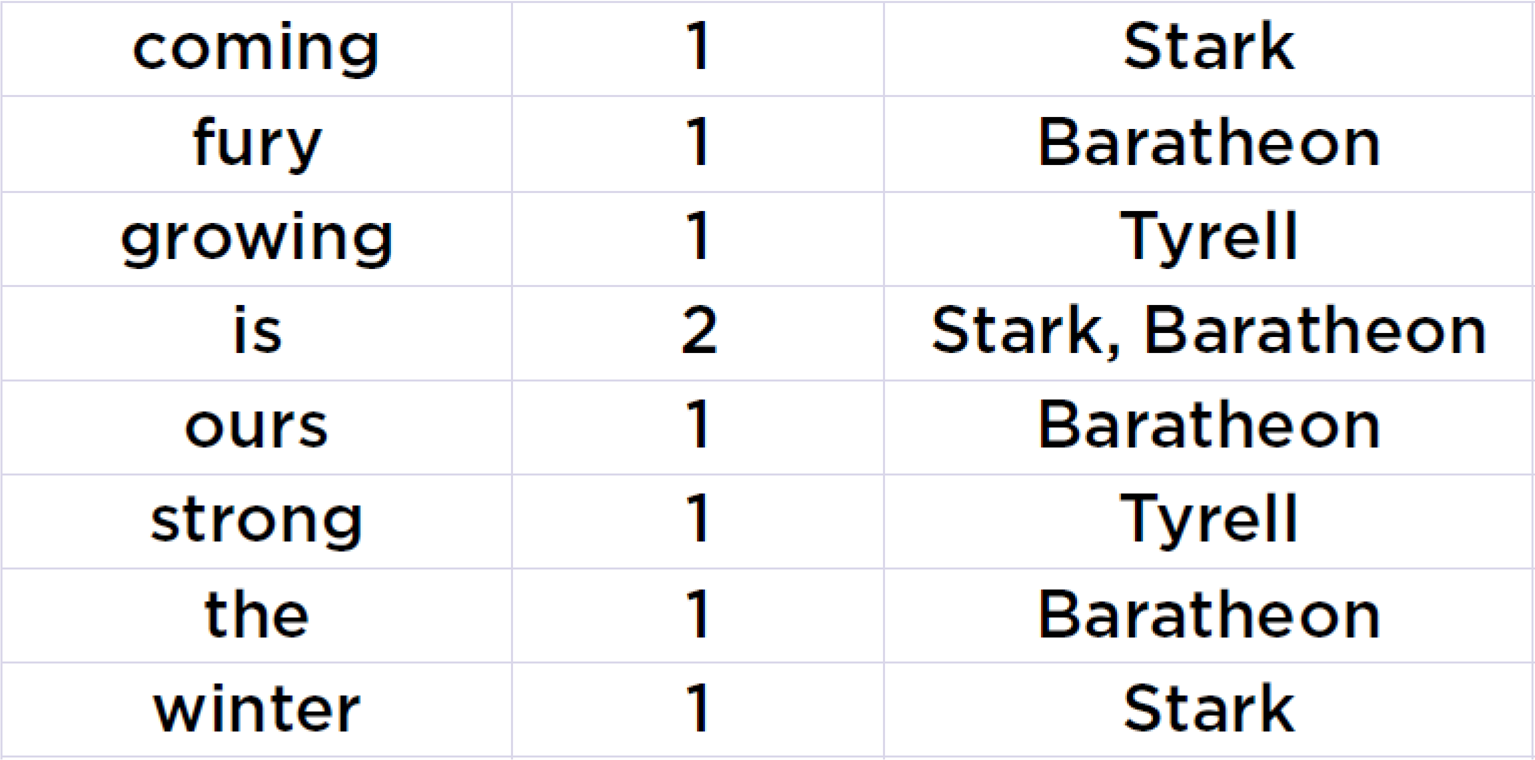


coming	1	Stark
fury	1	Baratheon
growing	1	Tyrell
is	2	Stark, Baratheon
ours	1	Baratheon
strong	1	Tyrell
the	1	Baratheon
winter	1	Stark

coming OR strong

Search

Inverted Index



coming	1	Stark
fury	1	Baratheon
growing	1	Tyrell
is	2	Stark, Baratheon
ours	1	Baratheon
strong	1	Tyrell
the	1	Baratheon
winter	1	Stark

fury and growing

Searches Using Inverted Indices

- Find all words ending with “ong”

strong → gnorts

- Search for all words starting with “gno”

Inverted Index

Searches Using Inverted Indices

- Split words into n-grams for substring search
- yours → yo, you, our, ours, urs
- Match substrings with n-grams

Inverted Index

Searches Using Inverted Indices

- Geo-hashes for geographical search
- Algorithms such as Metaphone for phonetic matching
- “Did you mean?” searches use a Levenshtein automaton

Inverted Index

- An inverted index is at the heart of a search engine

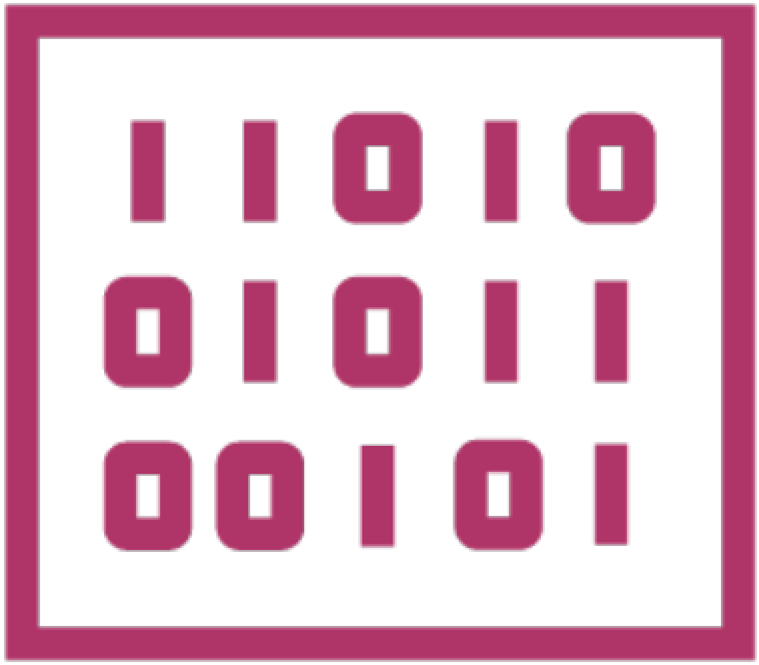
Implementing Search - Apache Lucene

Apache Lucene

The indexing and search library for a high performance, full-text search engine.

Open source, free to use written in Java, ported to other languages.

Just like Hadoop in the distributed computing world, Lucene is the nucleus of several technologies built around it.



Implementing Search - Apache Lucene

Apache Lucene

Solr

Web crawling and index parsing

Implementing Search - Apache Lucene

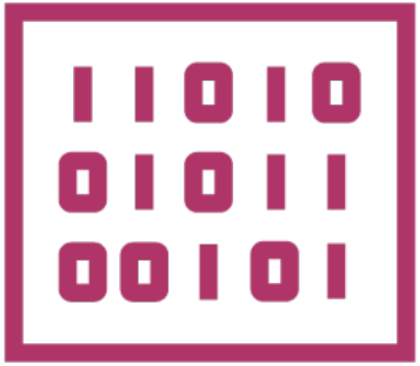
Apache Lucene

Nutch

Open source, free to use written in Java, ported to other languages

Implementing Search - Apache Lucene

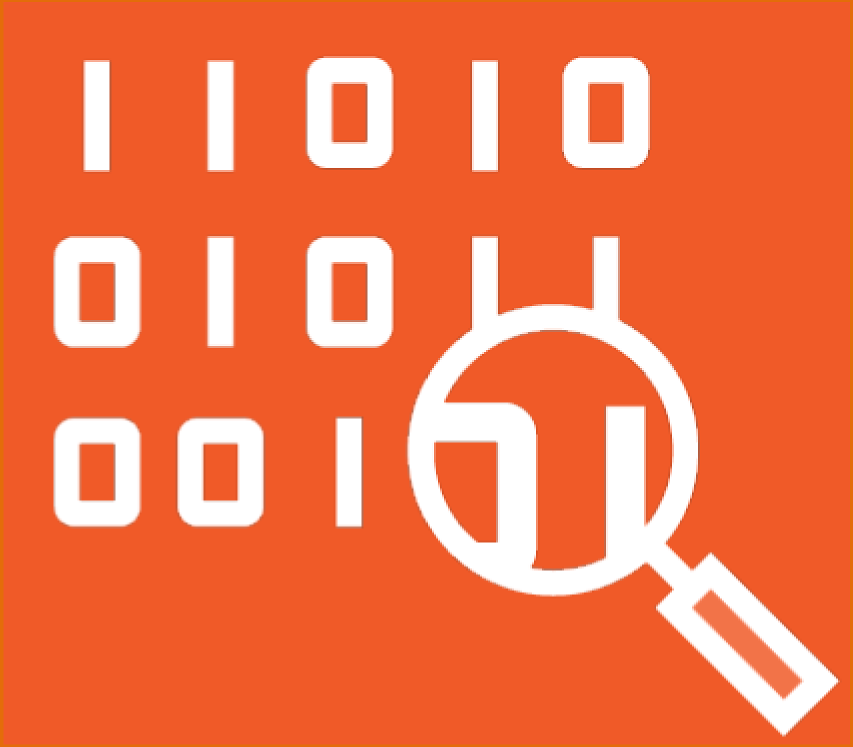
Apache Lucene



CrateDB

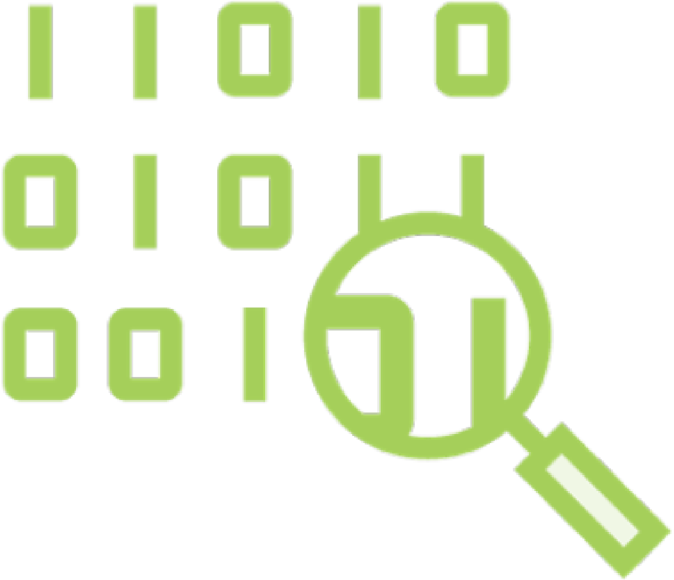
Open source, SQL distributed database

Elasticsearch



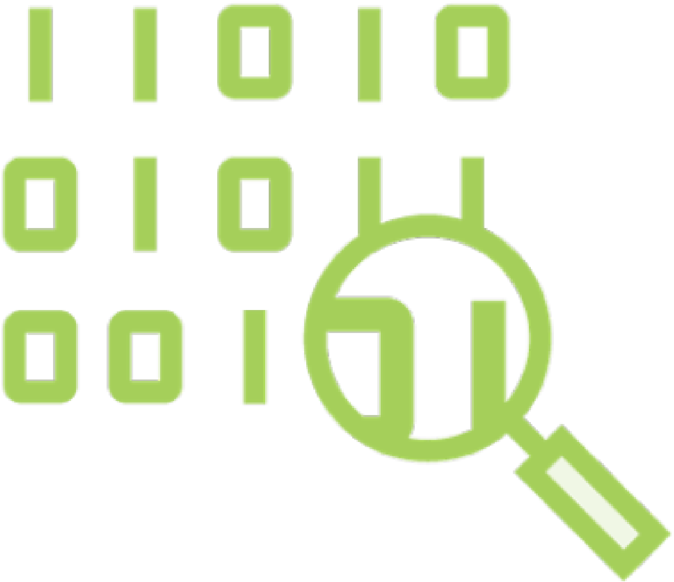
Elasticsearch is a distributed search and analytics engine which runs on Lucene

Introducing Elasticsearch



- An open source, search and analytics engine, written in Java built on Apache Lucene

Introducing Elasticsearch



- Distributed: Scales to thousands of nodes
- High availability: Multiple copies of data
- RESTful API: CRUD, monitoring and other operation via simple JSON-based HTTP calls
- Powerful Query DSL: Express complex queries simply
- Schemaless: Index data without an explicit schema

Elasticsearch

Product catalog

Inventory

Autocomplete

Video clips

Categories

Tags

Courses

Authors

Topics

Elasticsearch

Mining log data for insights

Price alerting platform

Business analytics and intelligence

Working with Elasticsearch

As a service in the cloud

On your local machine

https://www.elastic.co/cloud/as-a-service

Basic Concepts of Elasticsearch

Near Realtime Search

Very low latency, ~1 second from the time a document is indexed until it becomes searchable

Basic Concepts of Elasticsearch

Node

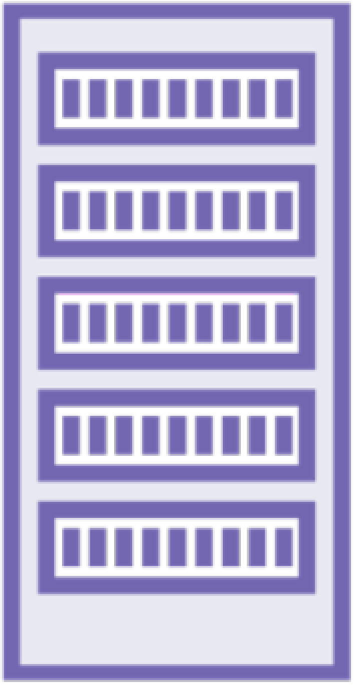
Single server

Performs indexing

Allows search

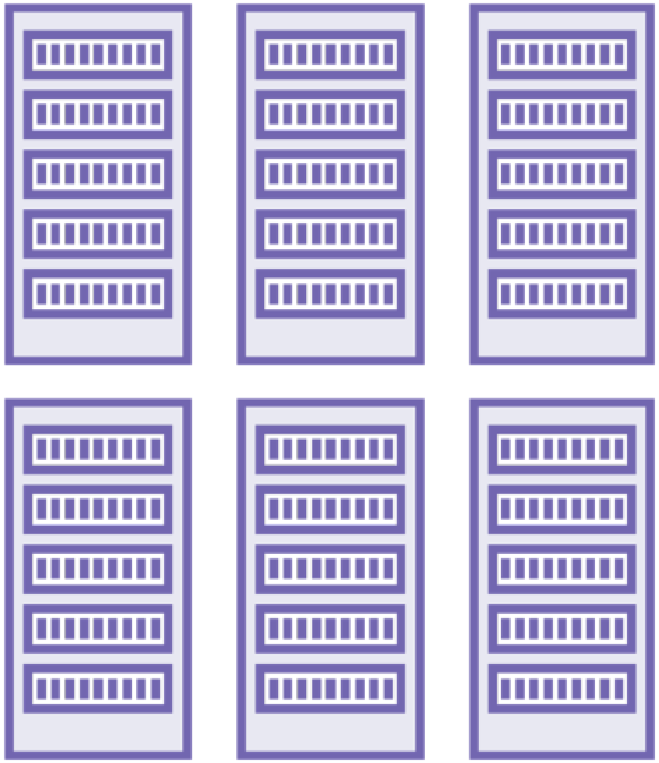
Has a unique id

and name



Basic Concepts of Elasticsearch

Cluster



Collection of nodes

Holds the entire indexed data

Has a unique name

Nodes join a cluster using the cluster name

Basic Concepts of Elasticsearch

Document

A whole bunch of documents that need to be indexed so they can be searched

Basic Concepts of Elasticsearch

Document

catalog , reviews

- titles, description, comments

Basic Concepts of Elasticsearch

Type

Documents are divided into categories or types

Basic Concepts of Elasticsearch

Index

All of these types of documents make up an index

Basic Concepts of Elasticsearch

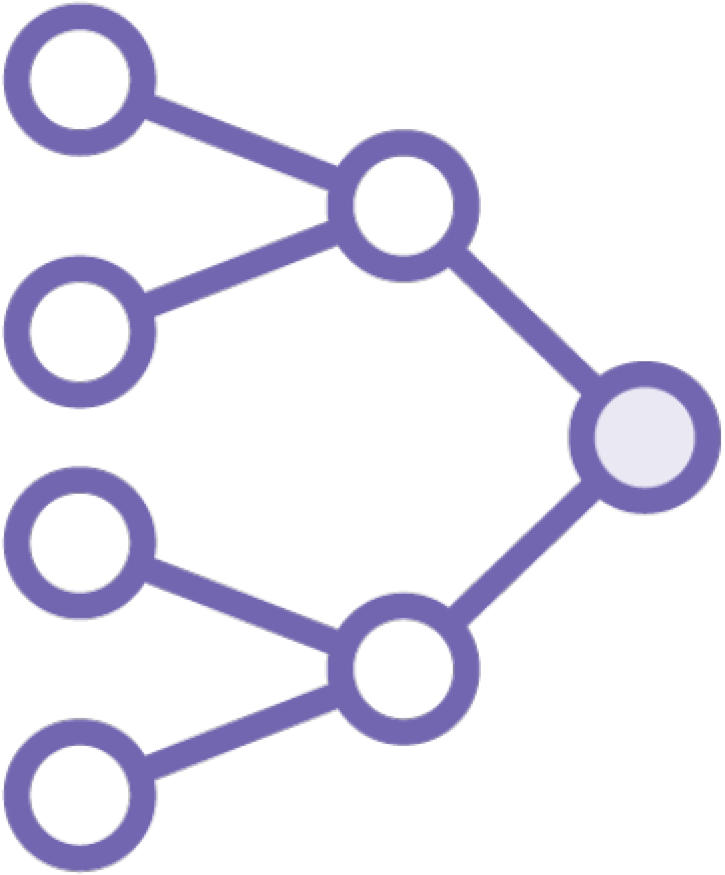
Index

Collection of similar documents

Identified by name

Any number of indices in a cluster

Different indices for different logical groupings



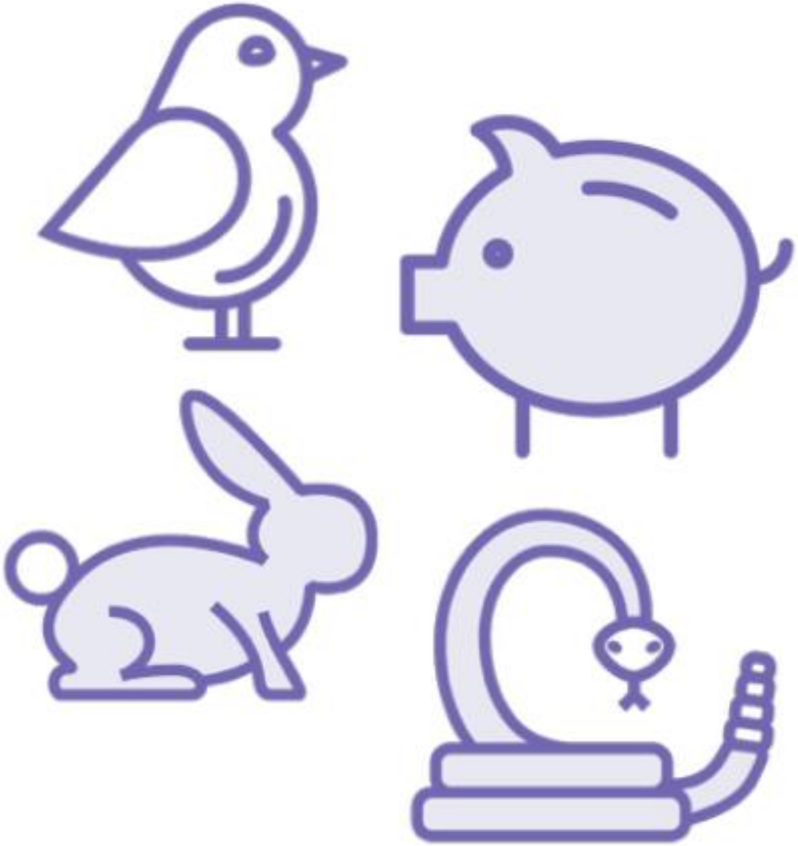
Basic Concepts of Elasticsearch

Type

Logical partitioning of documents

User defined grouping semantics

Documents with the same fields belong to one type



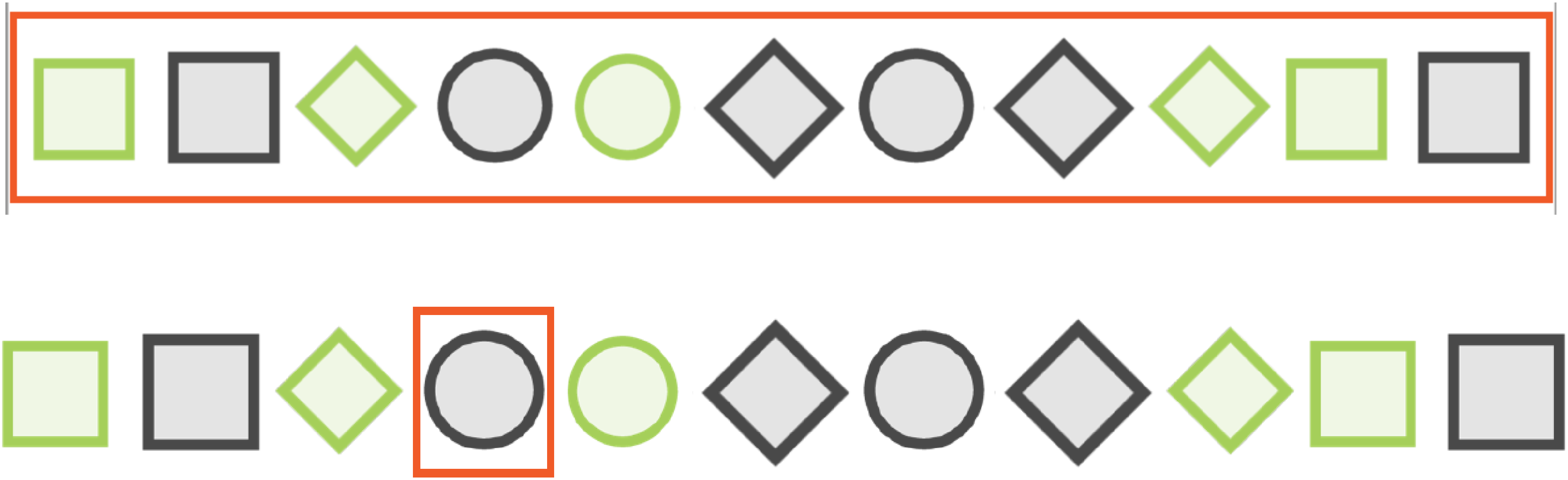
Basic Concepts of Elasticsearch

Document

- Basic unit of information to be indexed
- Expressed in JSON
- Resides within an index
- Assigned to a type within an index

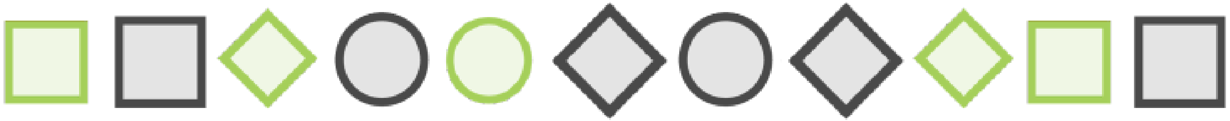
Index, Shards, Replicas

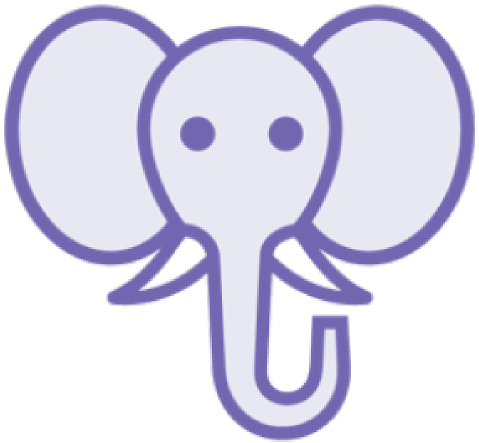
Document in an Index



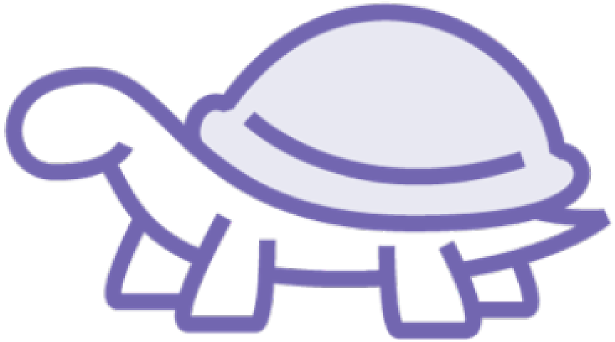
Index, Shards, Replicas

Document in an Index





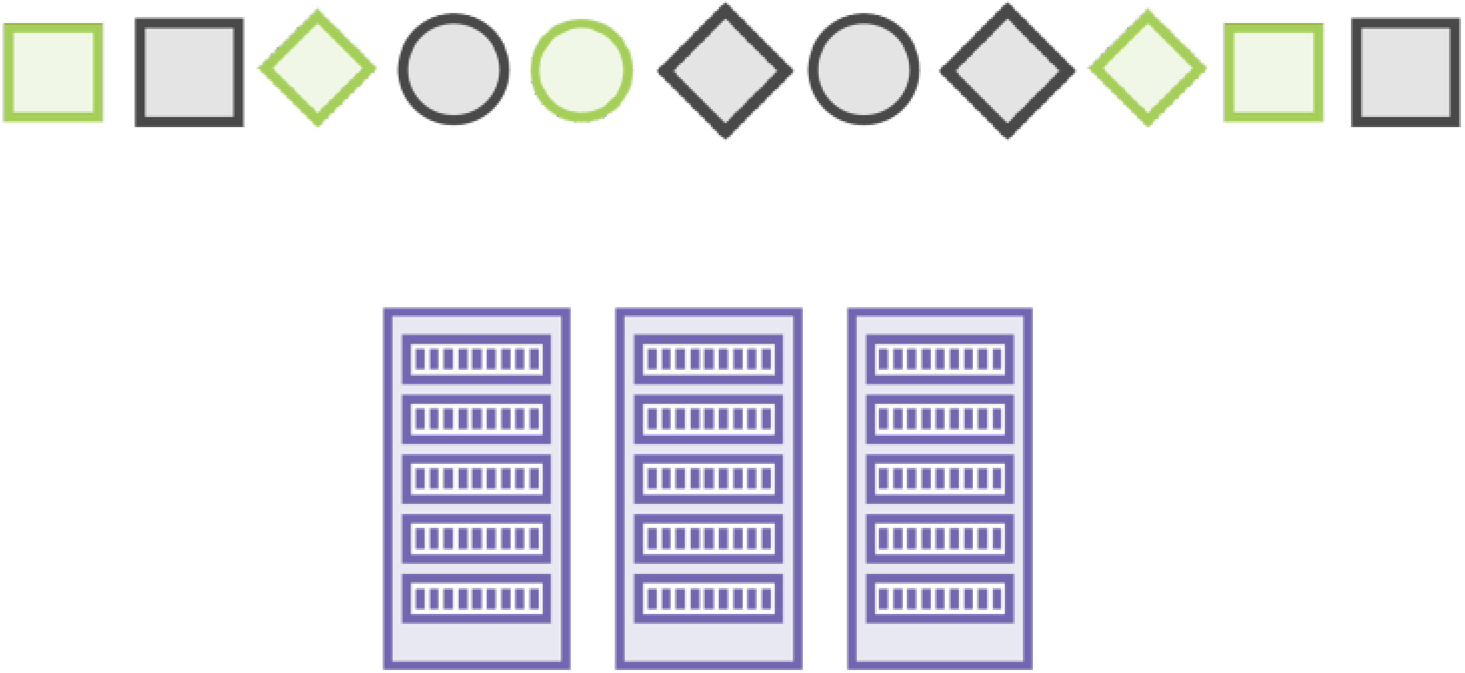
Too **large** to fit in the hard disk of one node



Too **slow** to serve all search requests from one node

Index, Shards, Replicas

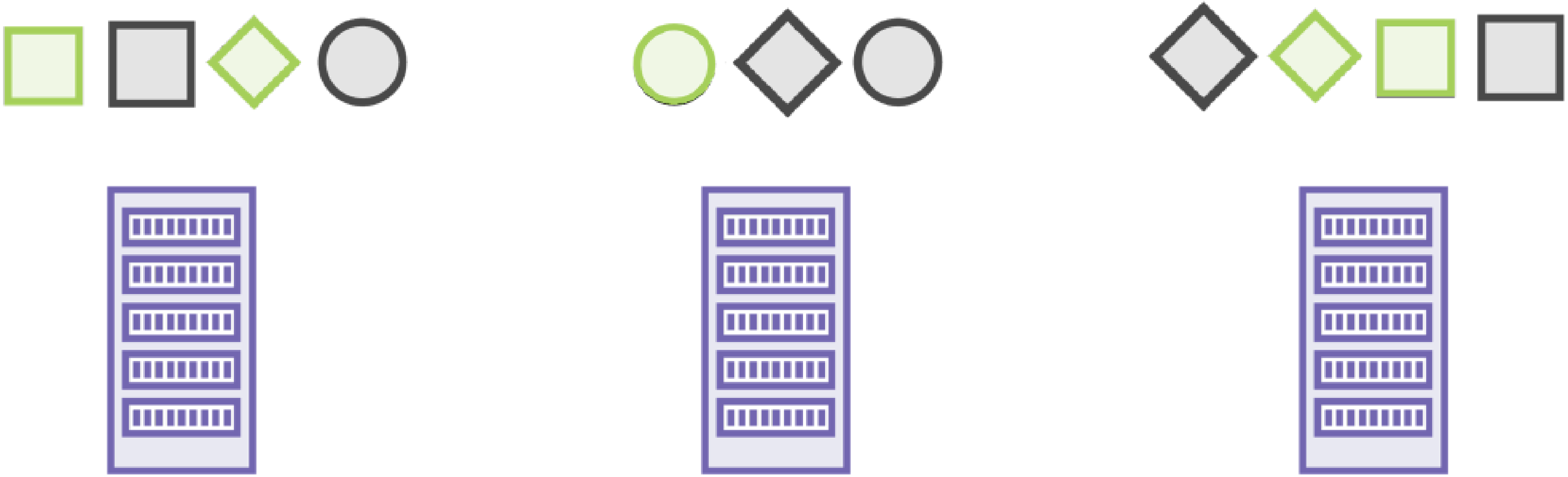
Shards



Split the index across multiple nodes in the cluster

Index, Shards, Replicas

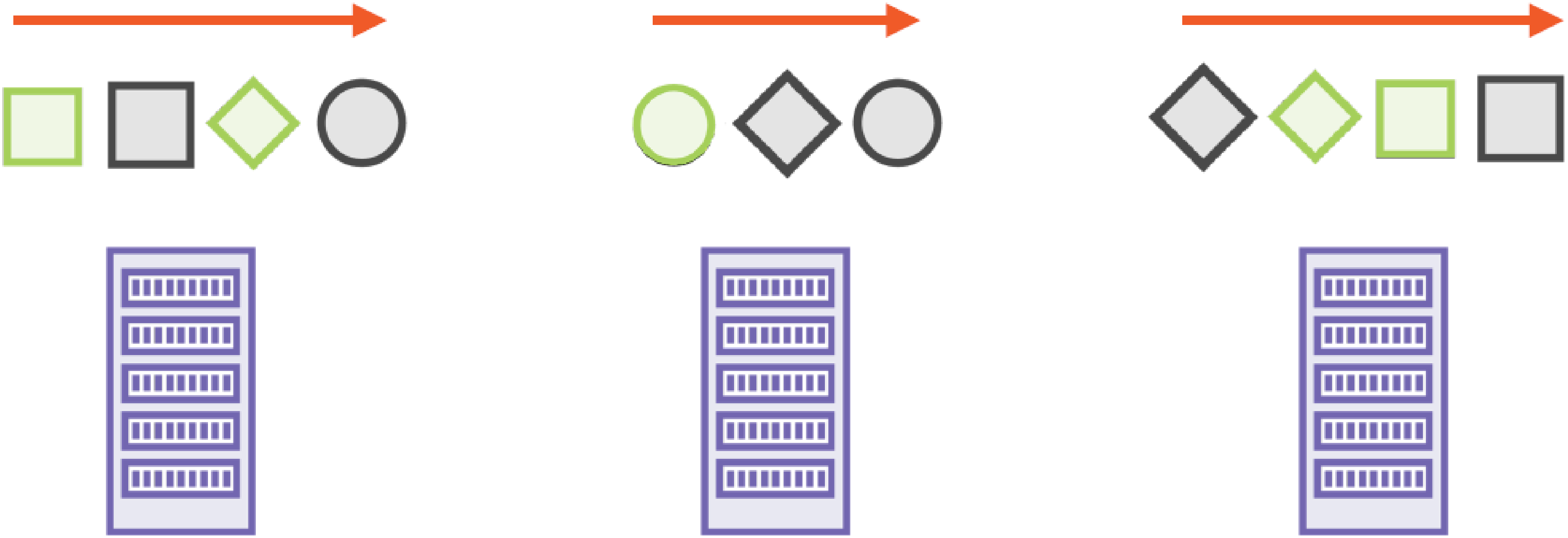
Shards



Sharding an index

Index, Shards, Replicas

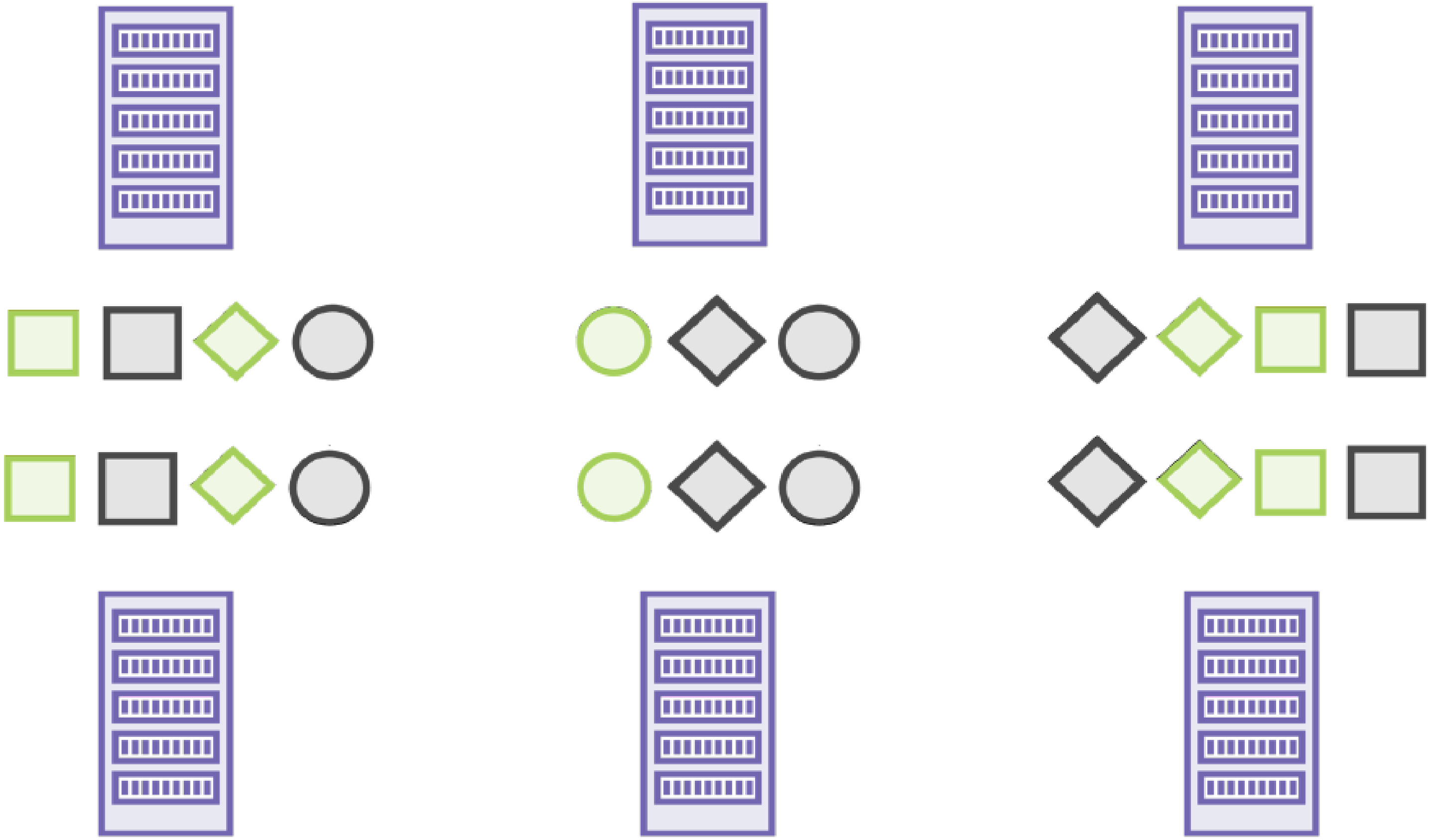
Shards



Search in parallel on multiple nodes

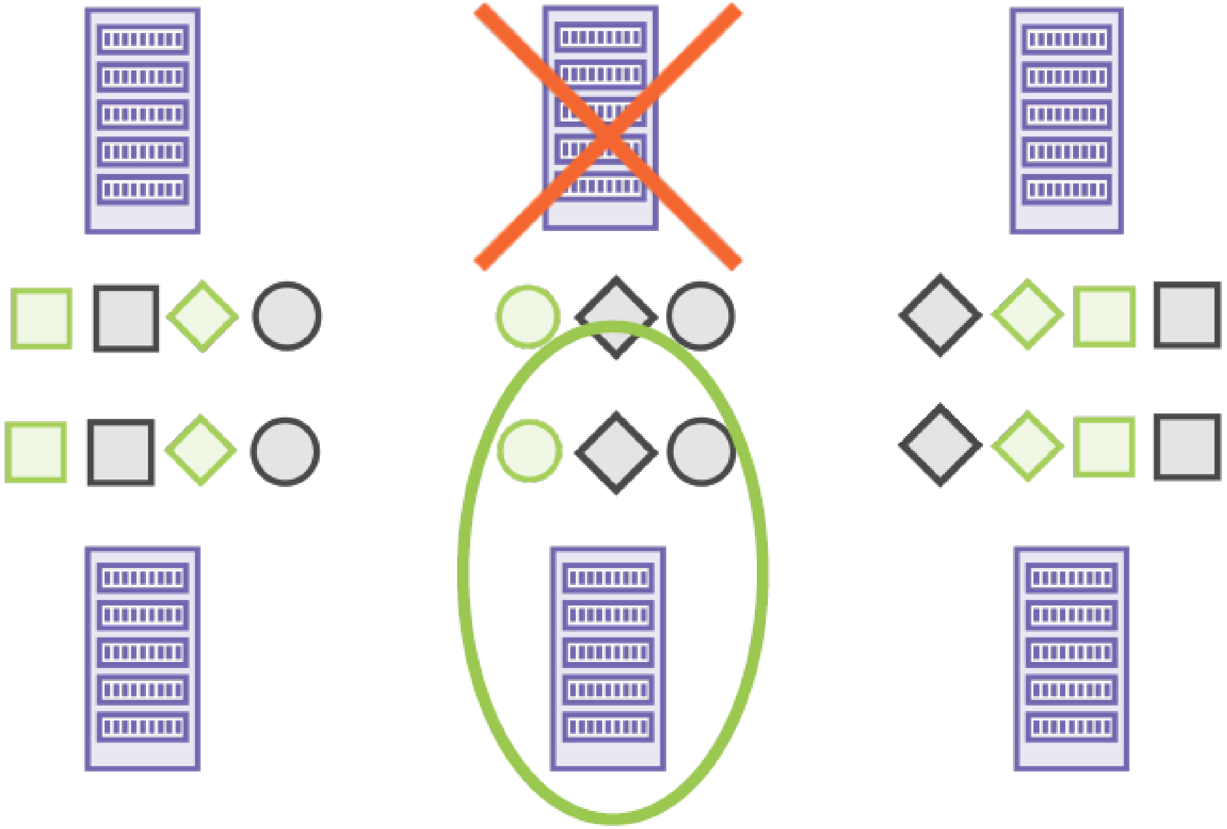
Index, Shards, Replicas

Replicas



Index, Shards, Replicas

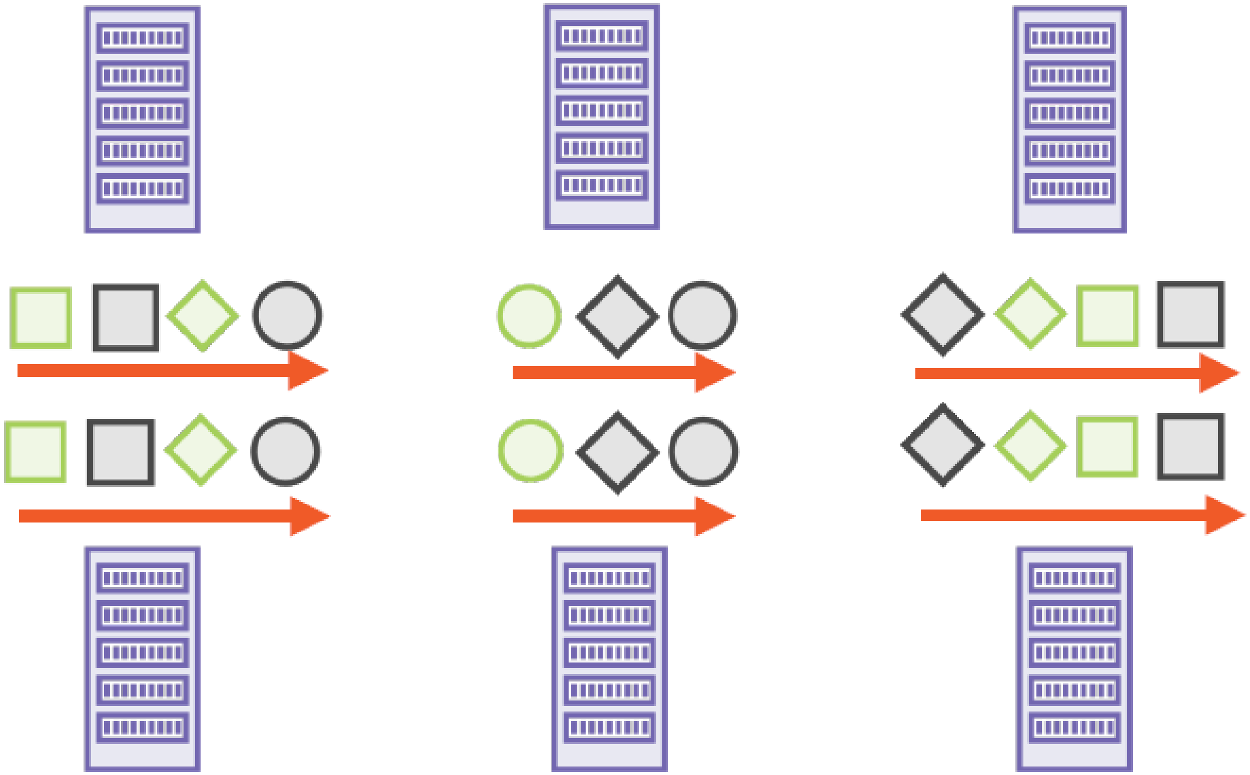
Replicas



High availability in case a node fails

Index, Shards, Replicas

Replicas

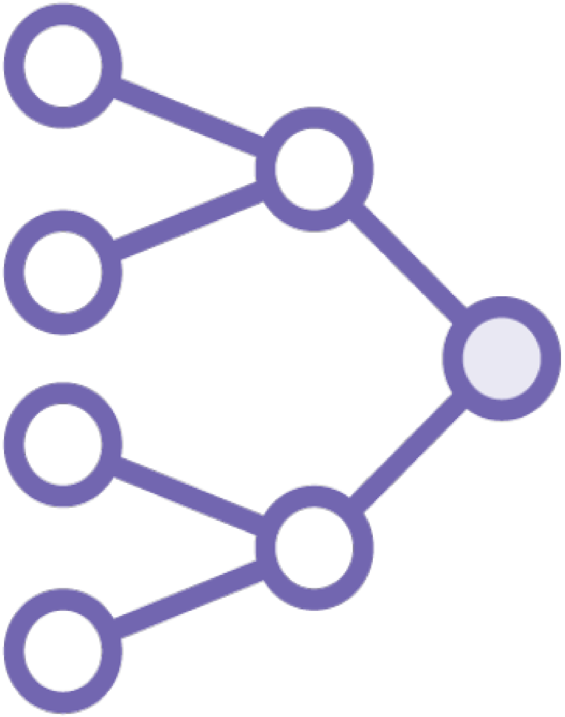


Scale search volume/throughput by searching multiple replicas

Index, Shards, Replicas

Shards and Replicas

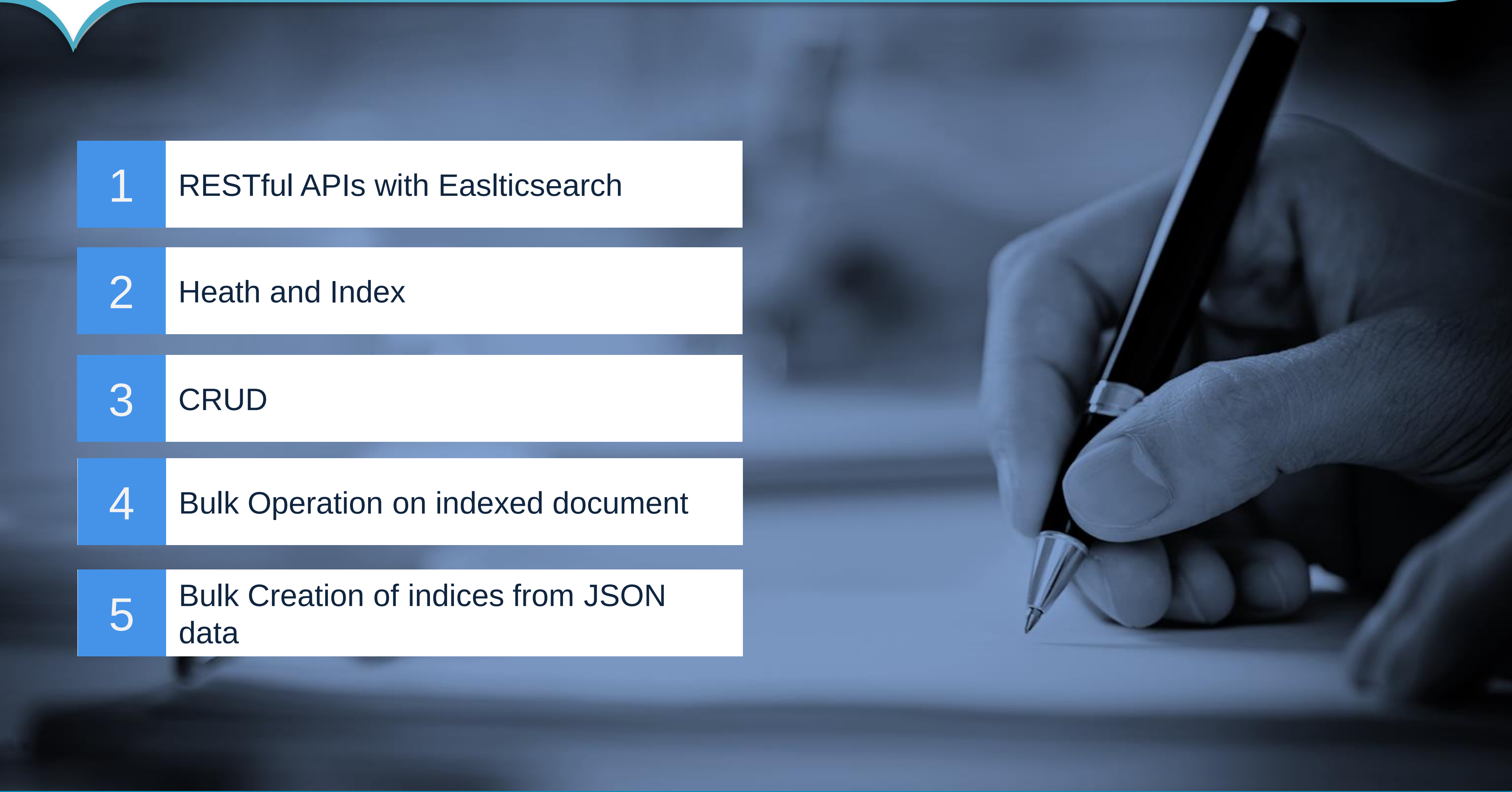
- An index can be split into multiple shards
- A shard can be replicated zero or more times
- An index in Elasticsearch has 5 shards and 1 replica by default



Summary

- Learnt a little search engine history, ubiquitous nature of search
- Understood the basics steps involved in indexing and searching documents
- Learnt how the inverted index data structure works
- Got a brief introduction to Elasticsearch and its building blocks
- Set up and installed Elasticsearch on your local machine

Course Map – CRUD operations using the Elasticsearch APIs



1. RESTful APIs with Easlticsearch
2. Heath and Index
3. CRUD
4. Bulk Operation on indexed document
5. Bulk Creation of indices from JSON data

RESTful APIs with Easlticsearch

RESTful APIs

- Elasticsearch uses REST APIs to administer the cluster, perform CRUD operations, search etc.
- Data is sent to and received from the server in JSON form

RESTful APIs with Easlticsearch

Cluster Health Status

```
curl "localhost:9200/_cat/health?v&pretty"
```

- **Green:**
  - All shards and replicas are available for requests, cluster fully functional
- Yellow:
  - Some replicas may not be available, cluster is still functional.
- Red:
  - Some shards not available, cluster NOT fully functional

RESTful APIs with Easlticsearch

cURL for Requests to REST APIs

- cURL is a tool which allows you to transfer data from and to a server using a variety of protocols
- HTTP, FTP, GOPHER, IMAP, LDAP etc.

CRUD

Demo

- Update documents by id:
  - whole documents
  - partial documents
- Delete documents in an index
- Delete the entire index

Bulk Operation on indexed document

Demo

- Bulk operations on documents:
  - – retrieve multiple documents
  - – index multiple documents
  - – multiple operations in one command

Bulk Creation of indices from JSON data

Demo

- Bulk index documents from a JSON file

Summary

- Performed CRUD operations on indexes holding documents
- Implemented bulk operations on indexed documents
- Created indices in bulk from JSON data in a file

Capgemini

CONSULTING.TECHNOLOGY.OUTSOURCING

People matter, results count.

About Capgemini

With almost 180,000 people in over 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2014 global revenues of EUR 10.573 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

www.capgemini.com