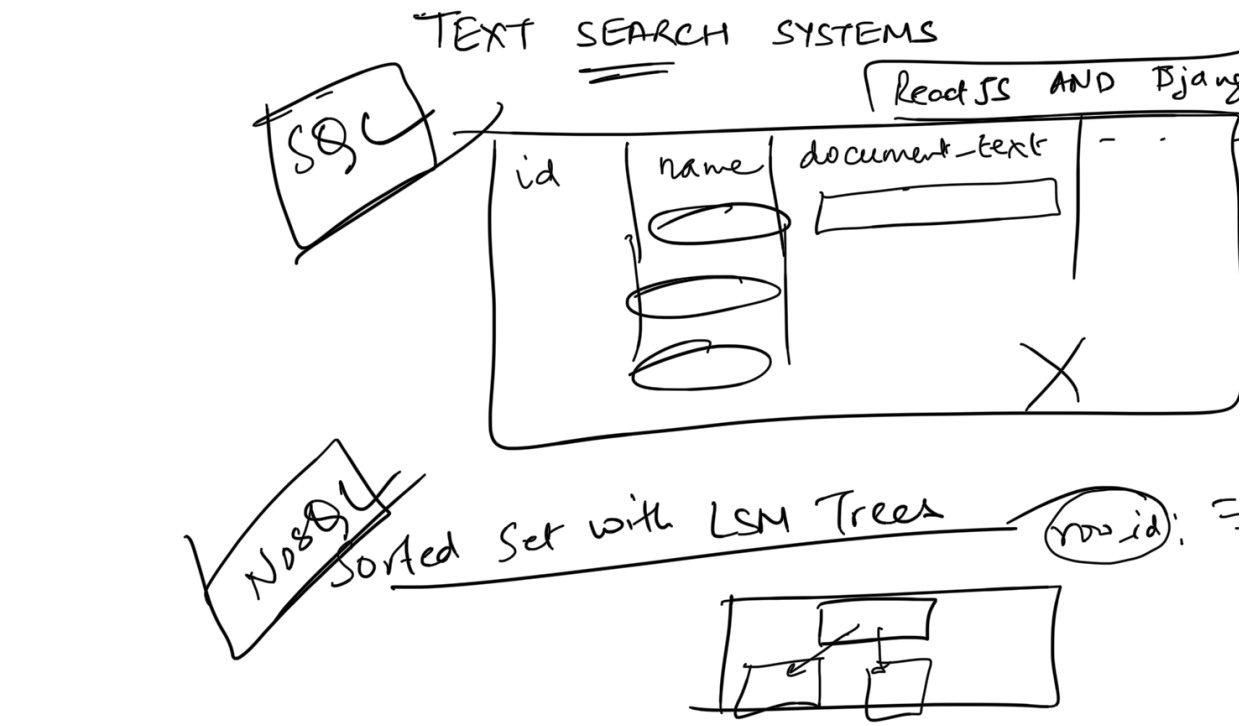
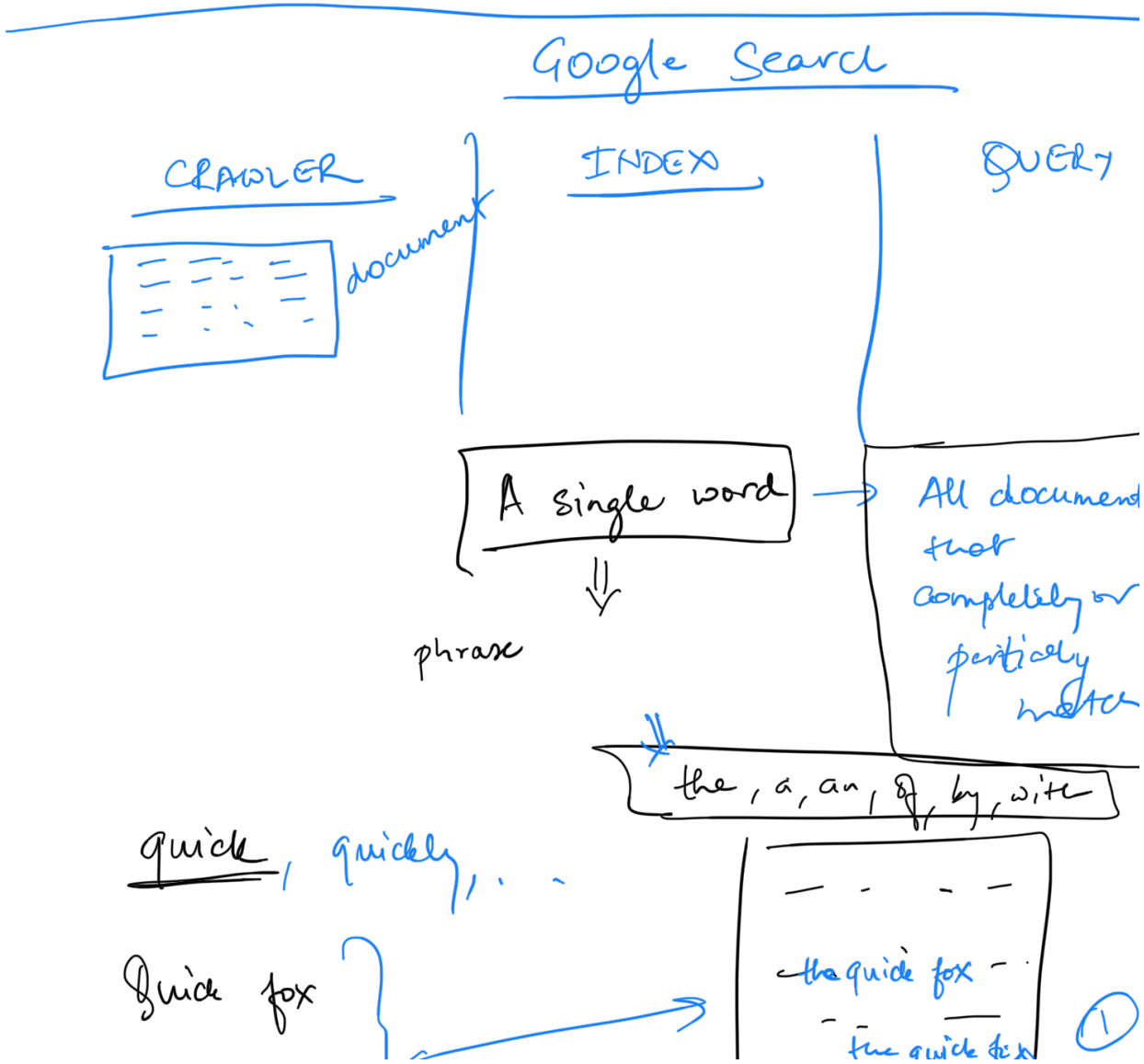
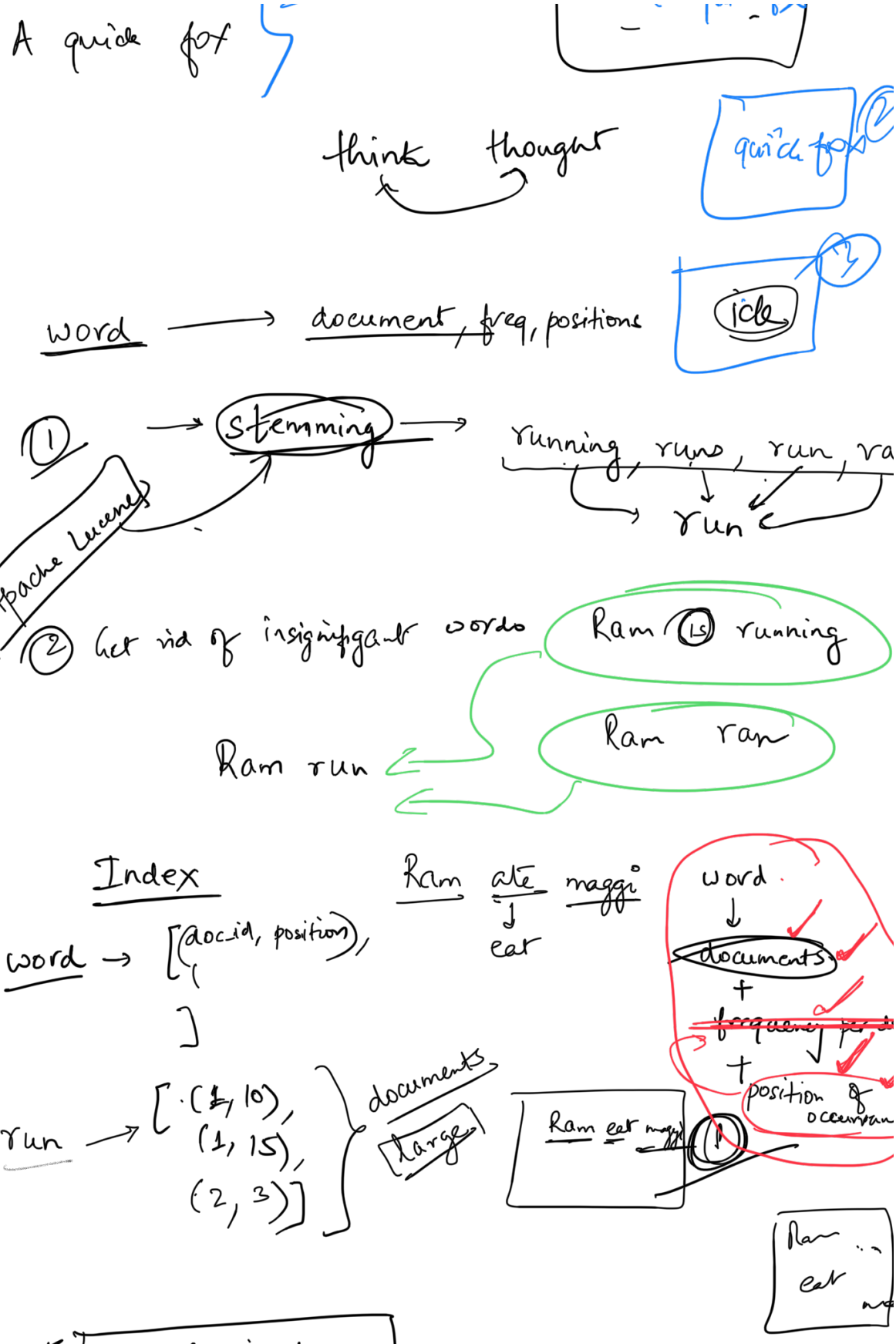
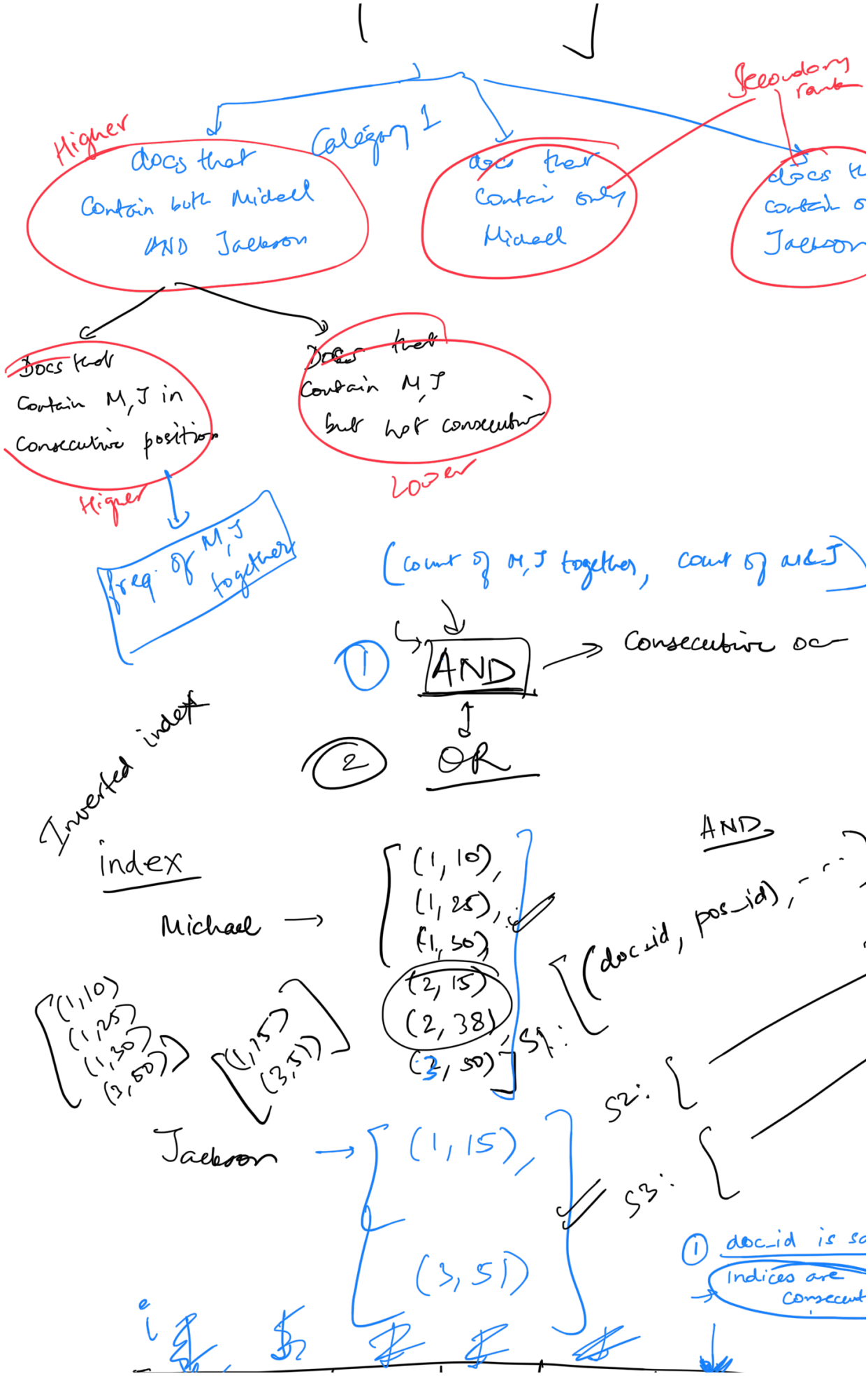
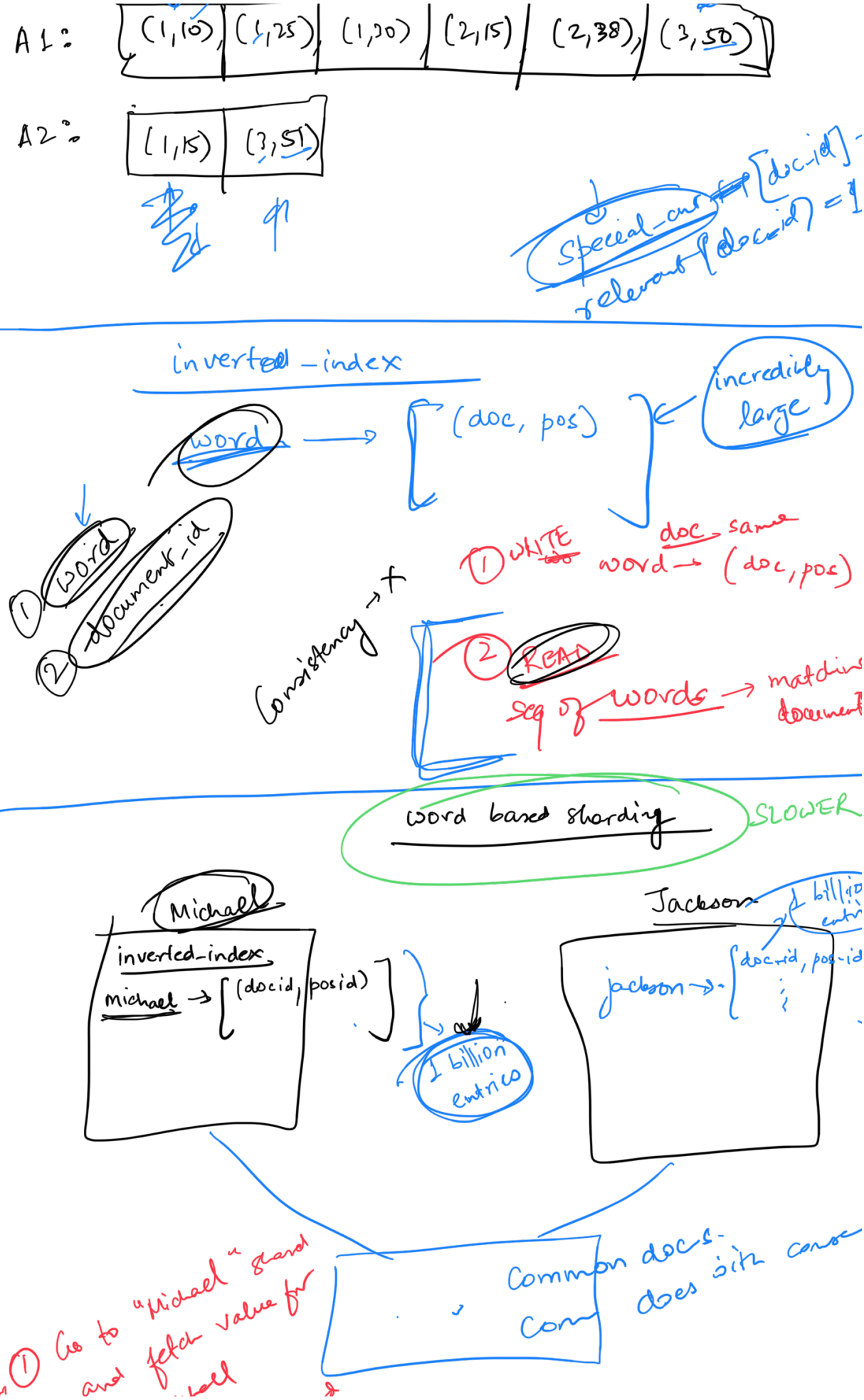
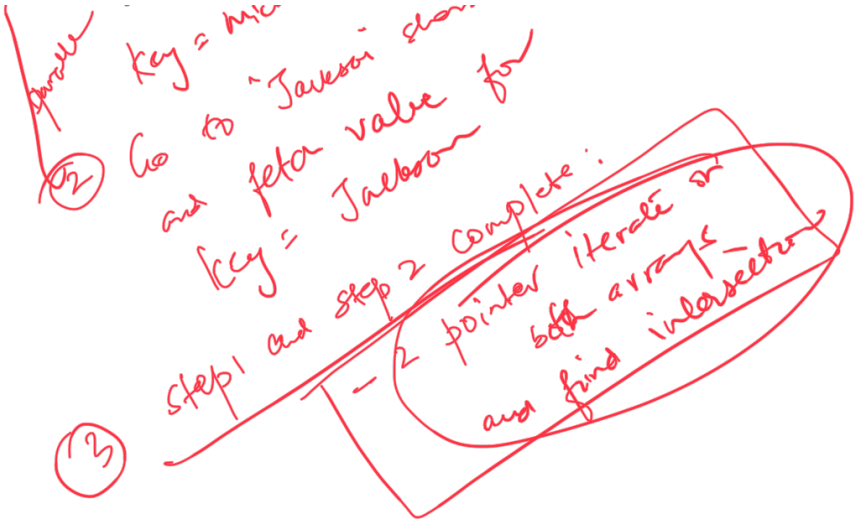

TEXT SEARCH SYSTEMS
SQL
ReactJS AND Djang
id
name
document_text
NoSQL
Sorted Set with LSM Trees
row_id:

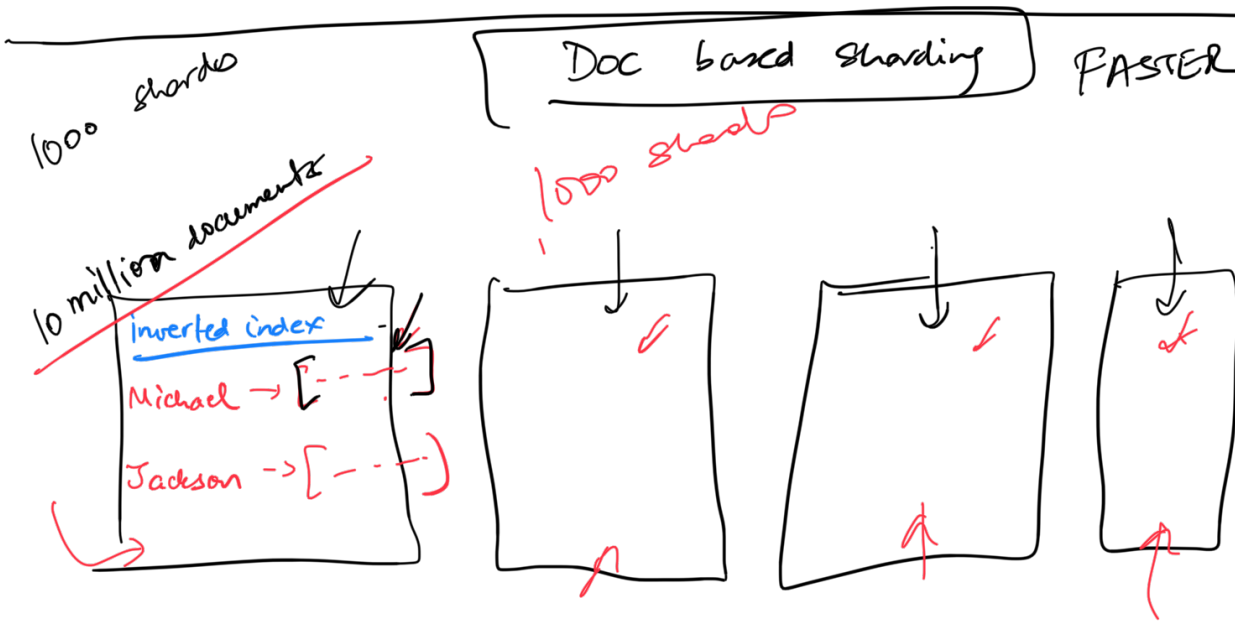

Google Search
CRAWLER
INDEX
QUERY
document
A single word
All document
that
completely or
partially
match
phrase
the, a, an, of, by, with
quick, quickly, . . .
quick fox
the quick fox
the quick fox
1


A quick for
think thought
quick for
word → document, freq, positions
icle
Apache Lucene
1
stemming
running runs run ra
run
2 Get rid of insignificant words
Ram is running
Ram ran
Ram run
Index
word → [(doc_id, position),]
Ram ate maggi
eat
word
documents
+
+
position of occurrence
run → [(1, 10), (1, 15), (2, 3)]
documents
large
Ram eat maggi
Ram eat
Michael jackson
Index
word → (doc_id, pos_id)


Higher
Category 1
docs that
Contain both Michael
AND Jackson
Docs that
Contain only
Michael
Jackson
Docs that
Contain M, J in
Consecutive position
Higher
Docs that
Contain M, J
but not consecutive
Lower
Freq of M, J together
(count of M, J together, count of all J
1
AND
Consecutive oc
2
OR
Inverted index
index
Michael →
(1, 10),
(1, 25),
(1, 50)
(2, 15)
(2, 38)
(2, 50)
S1:
(doc_id, pos_id),
AND
(1,10)
(1,25)
(1,50)
(2,50)
(1,15)
(3,51)
Jackson →
(1, 15),
(3, 51)
S2:
S3:
1 doc_id is so
Indices are
consecut

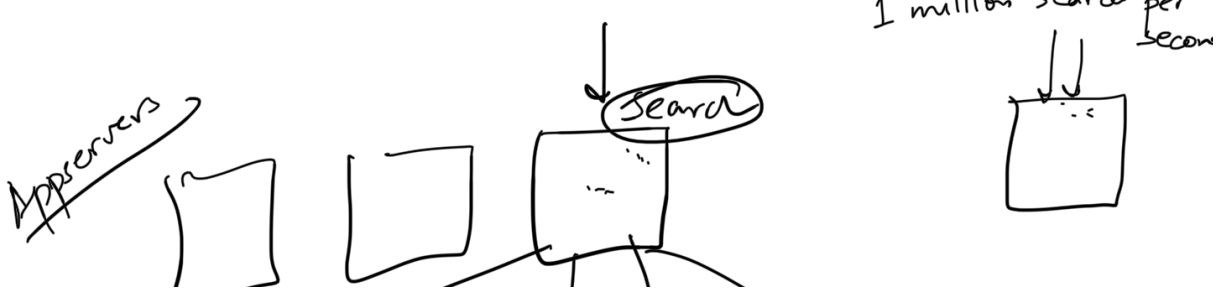

A1:
(1,10) (1,25) (1,30) (2,15) (2,38) (3,50)
A2:
(1,15) (3,51)
inverted-index
word
(doc, pos)
incredibly large
document_id
consistency
WRITE
doc same
word → (doc, pos)
READ
seq of words → matching
word based sharding
SLOWER
Michael
inverted-index
Michael → [(docid, posid)]
1 billion entries
Jackson
jackson →
Common docs
does with conse
Go to "Michael" shard and fetch value for



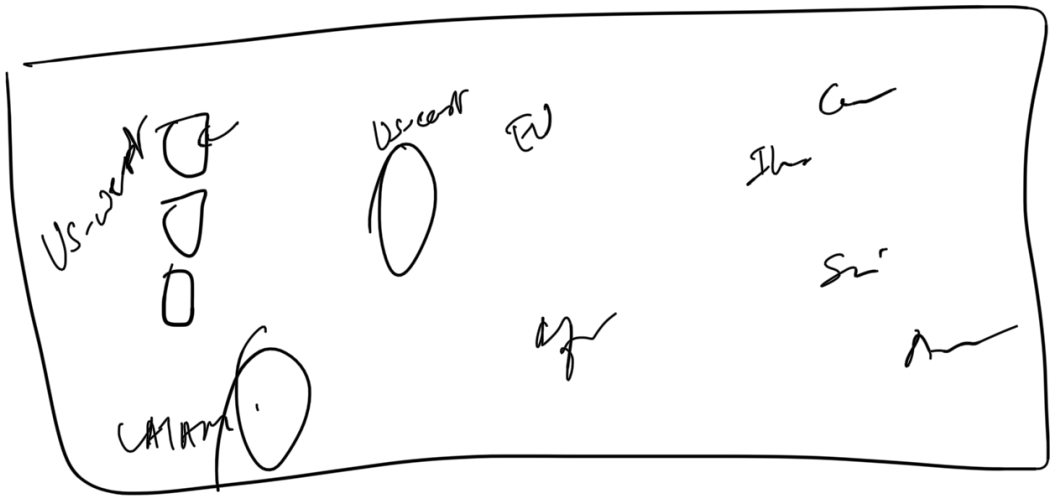


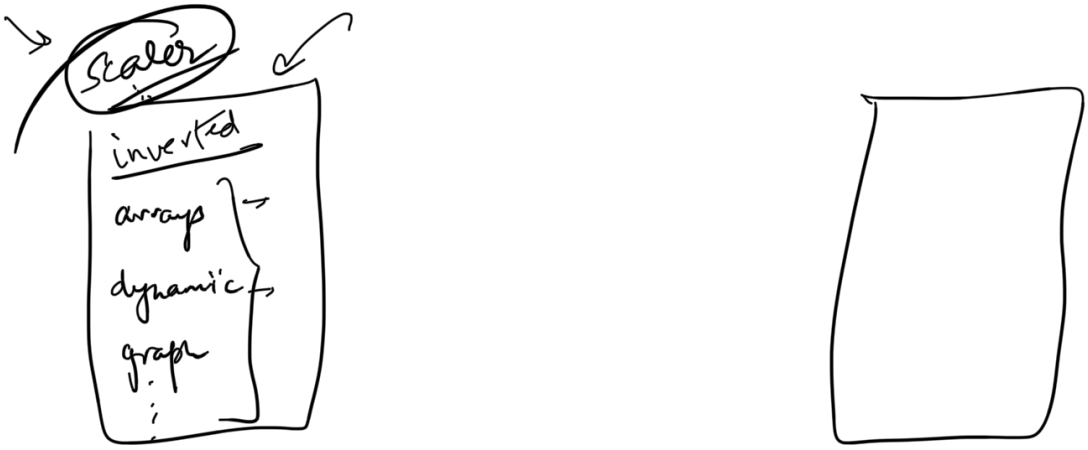
(1) In every shard, in parallel, 0.1-0.2 se
2 pointer iterate on value arr
and find intersection

(2) Wait for shards 0.5 second
→ (doc-id, X, Y, Z)

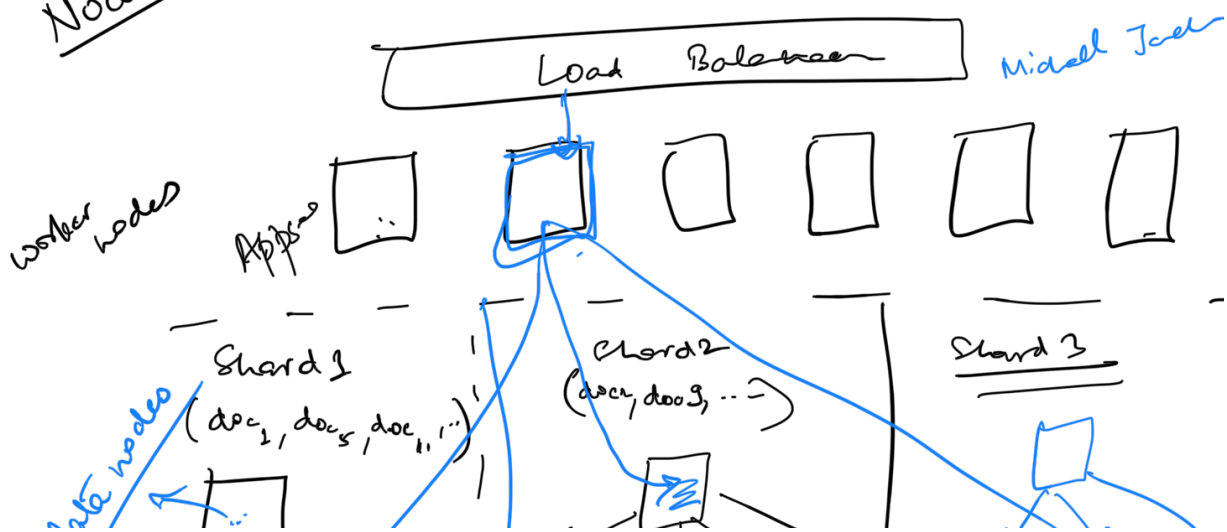


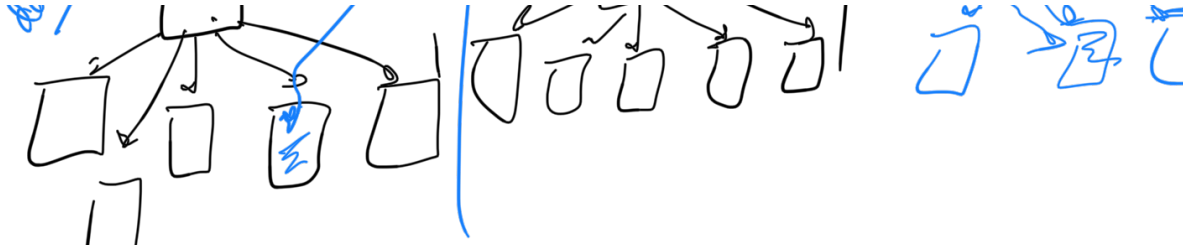

1000 sh




Scaler
inverted
arrays
dynamic
graph

Nodes


Load Balancer
Shard 1
Shard 2
Shard 3



Shards are independent

ES → Apache Lucene
- stemming
- inverted index

Google Seard
- Keyword mated
  - consecutive
  - freque
  - all vs any
- Personalisation
- Page Ranking
  - Domain auth
  - Page score
- Typo

User → resumes

Keywords → resum
users

Text search

" Puma Men XL Pullover NextGen White "

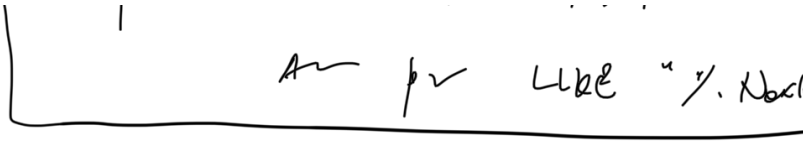
Pullover NextGen

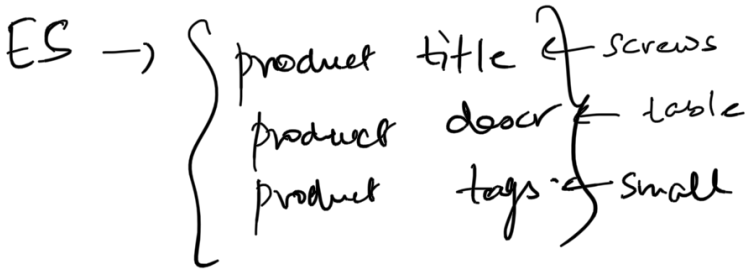
Puma NextGen

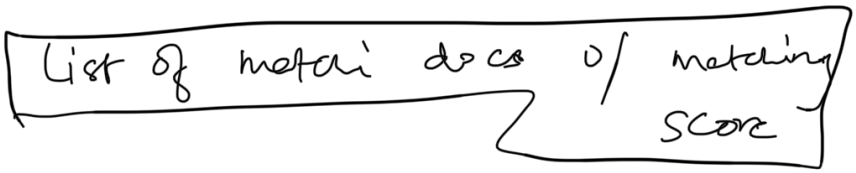
Pullover white

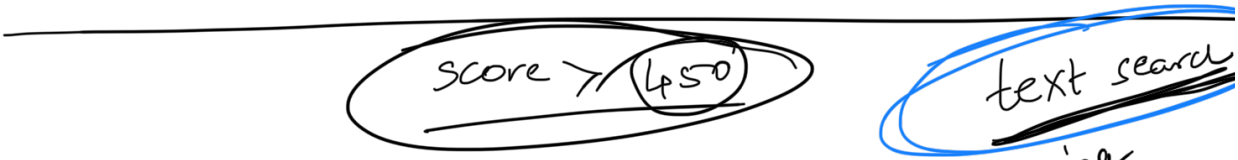
text Se

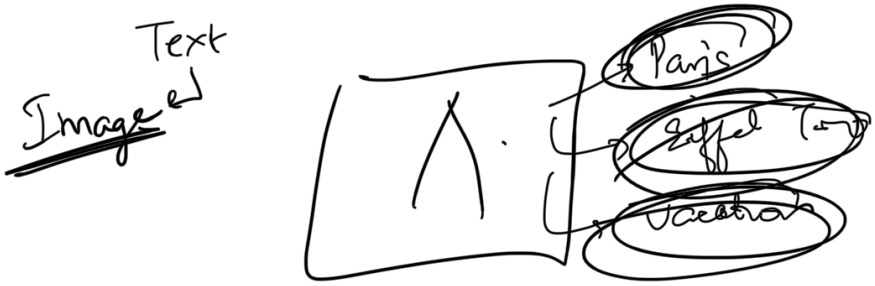
product_name LIKE "%Pullover%"


Ans for like "%.Nac


ES →
product title
product descr
product tags
screws
table
small


List of match docs w/ matching score


score > 450
text search
string


Text
Image
Paris
Eiffel Tower
Vacation

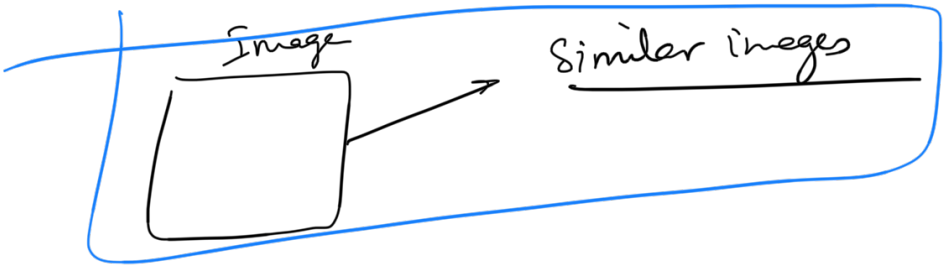
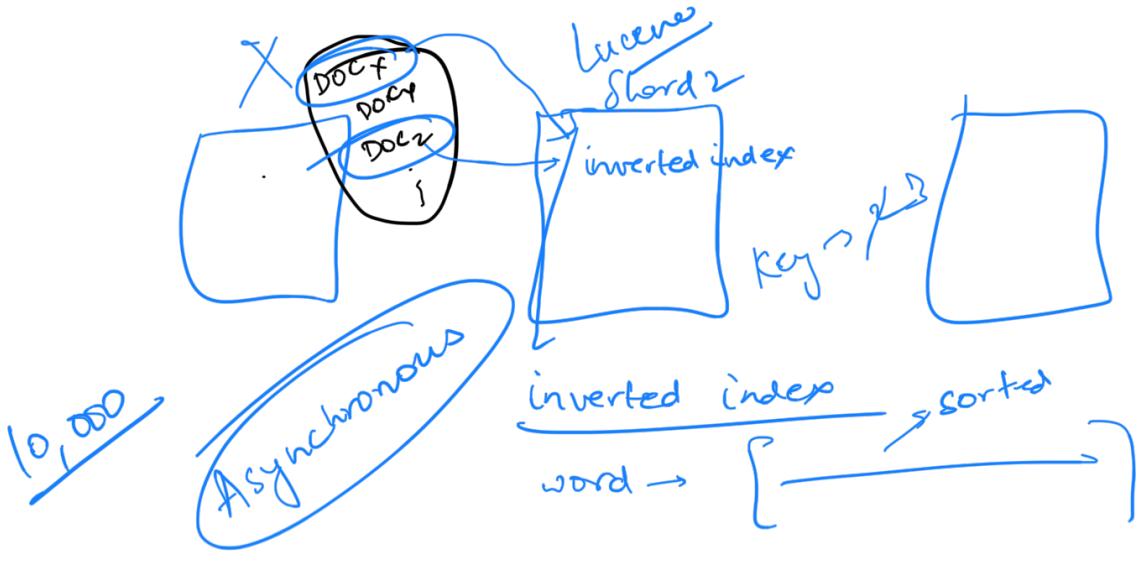


Image
Similar images


X
Doc1
Doc4
Doc2
Lucene
Solr
inverted index
key → K
10,000
Asynchronous
inverted index
sorted
word →