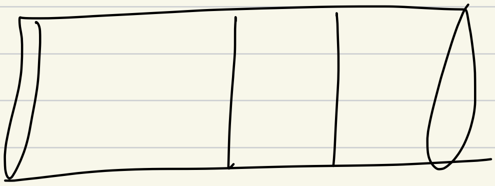
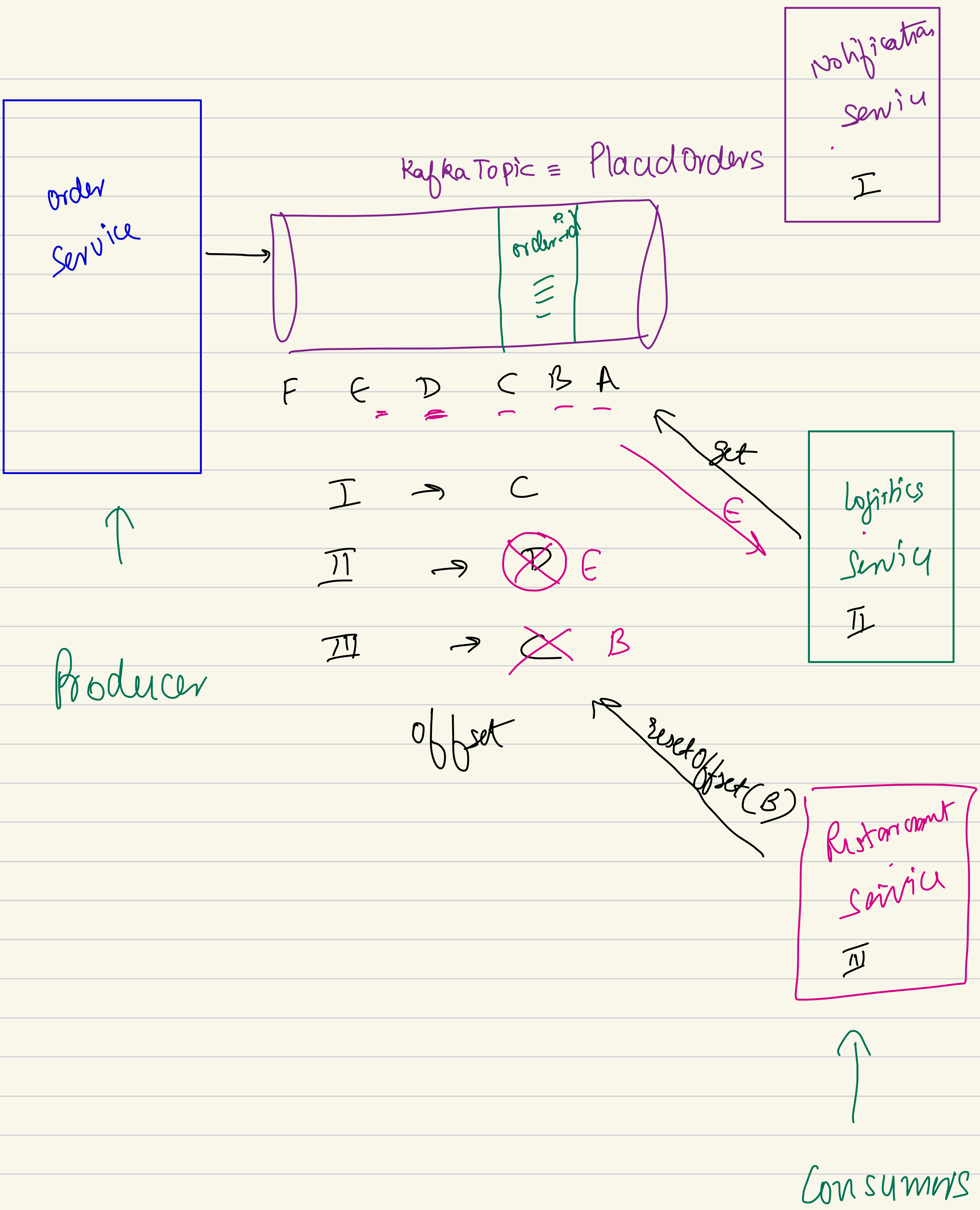
18 | Dec | 2023

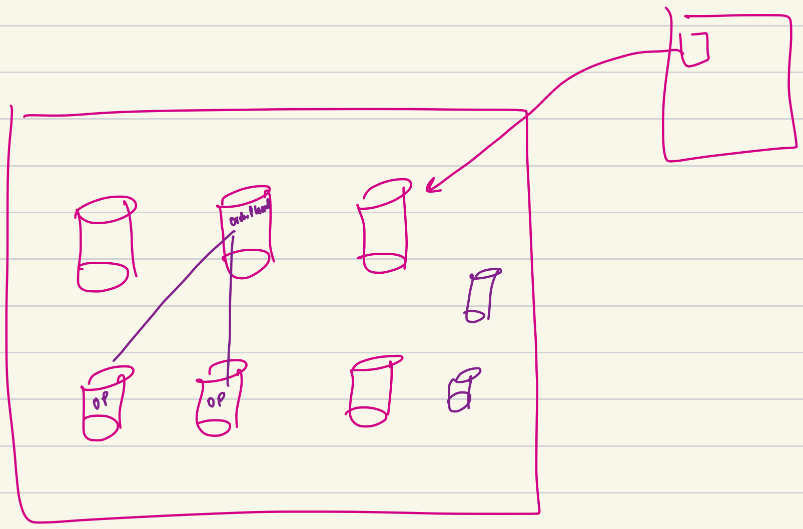
1. Kafka — Partitions
2. Full Text Search
3. NLP Techniques
4. Elastic Search

Swiggy





Order Service
Kafka Topic ≡ PlacedOrders
P.K.
order-id
F E D C B A
I → C
II → ~~D~~ E
III → ~~C~~ B
offset
get
E
resetOffset(B)
Notification Service I
Logistics Service II
Restaurant Service IV
Producer
Consumers



Given that I want to build Kafka as a dependable / fault tolerant system, the fundamental concept of replication should be employed

There may be a situation that 1 entire machine may NOT be sufficient to store all msgs of 1 queue also.

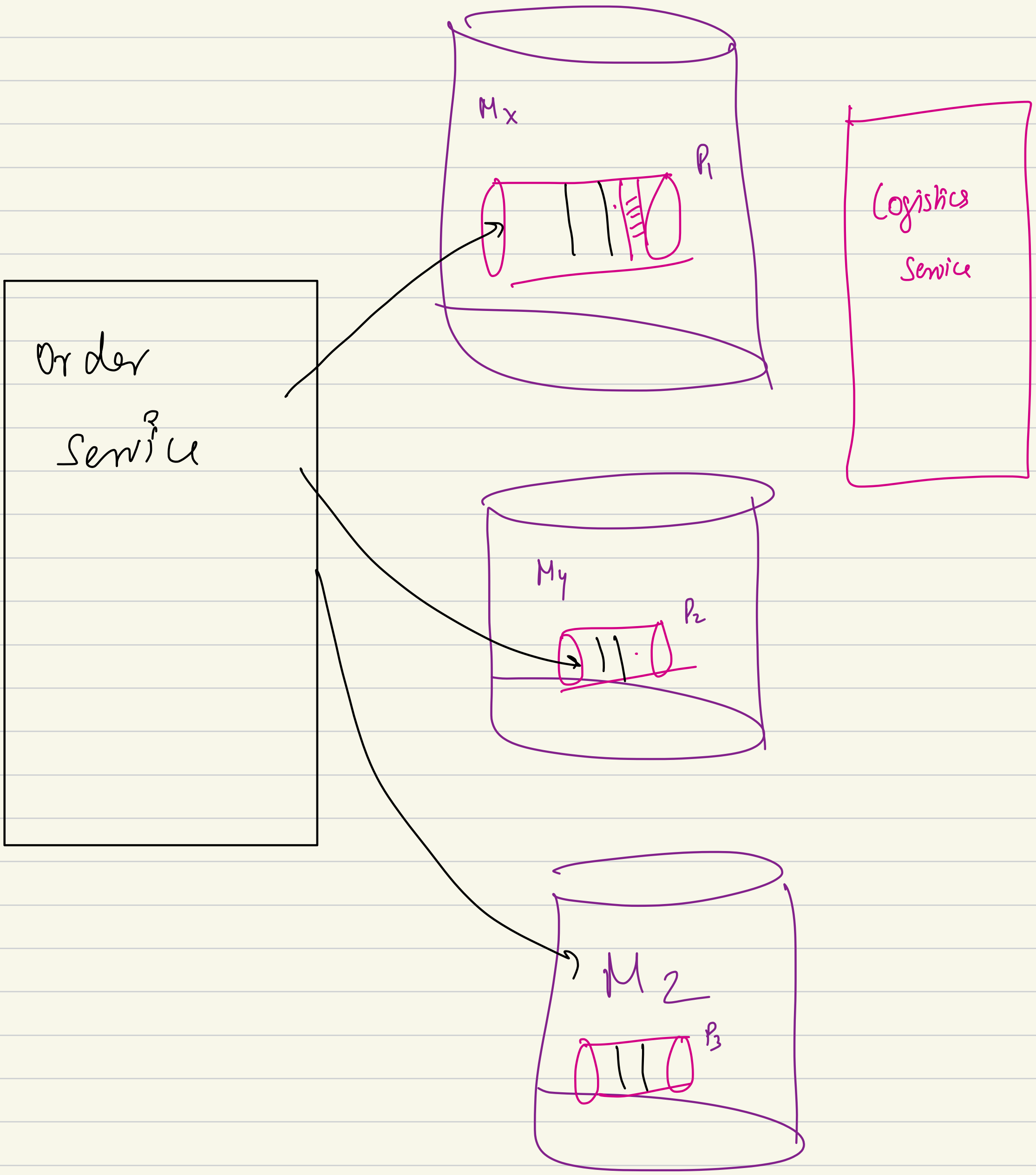
The usual way to solve this → SHARDING

But we can't use sharding directly before understanding it...

Queue is FIFO

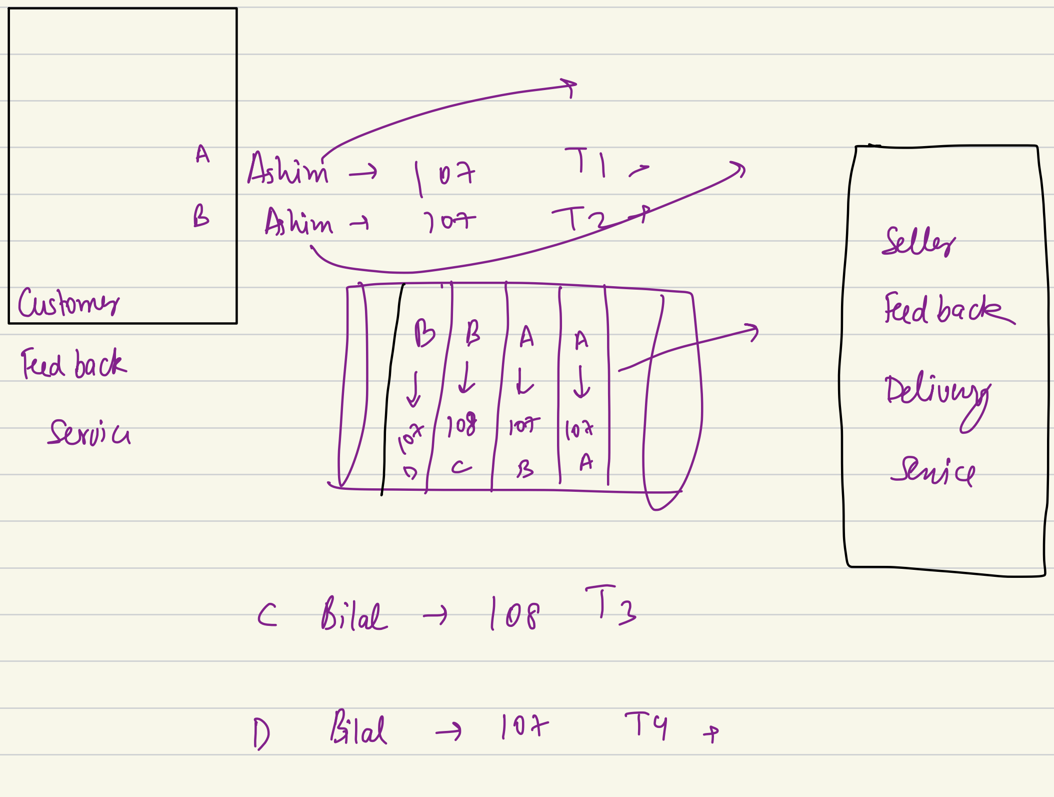
PROBLEM: When you divide a queue across multiple machines, it No longer remains 1 queue; it becomes multiple pieces of 1 queue.

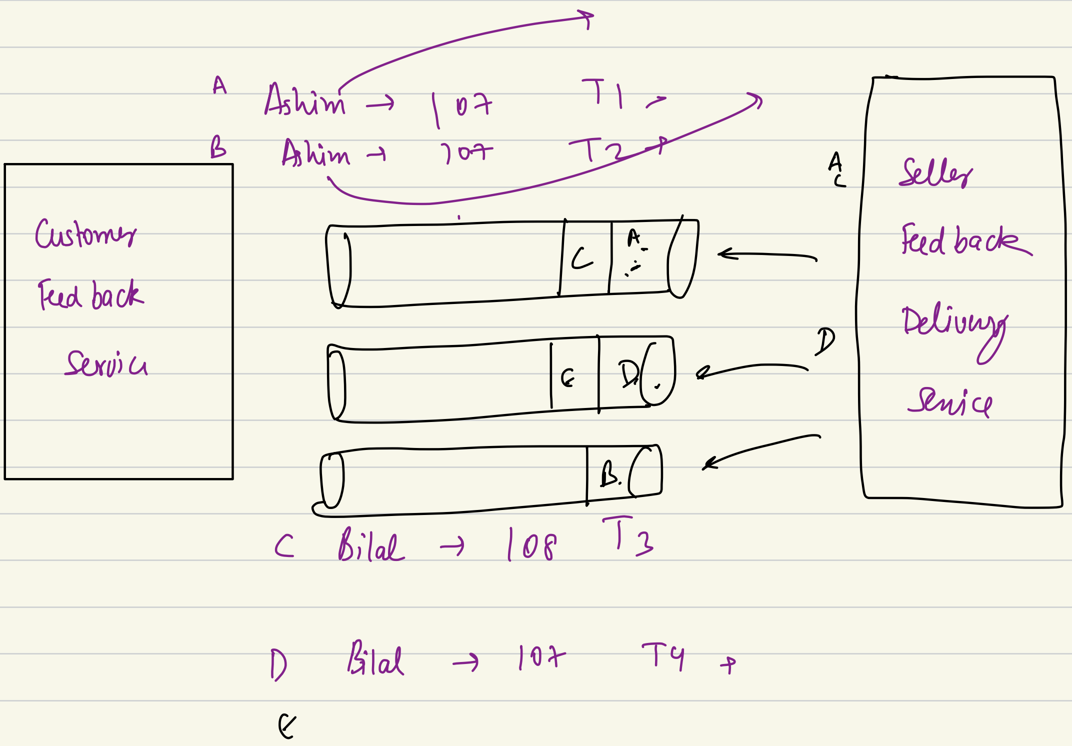
Kafka Topic | Queue ≡ Placed Orders



The Kafka Topic when partitioned, no longer gives FIFO gaurantees across all partitions.

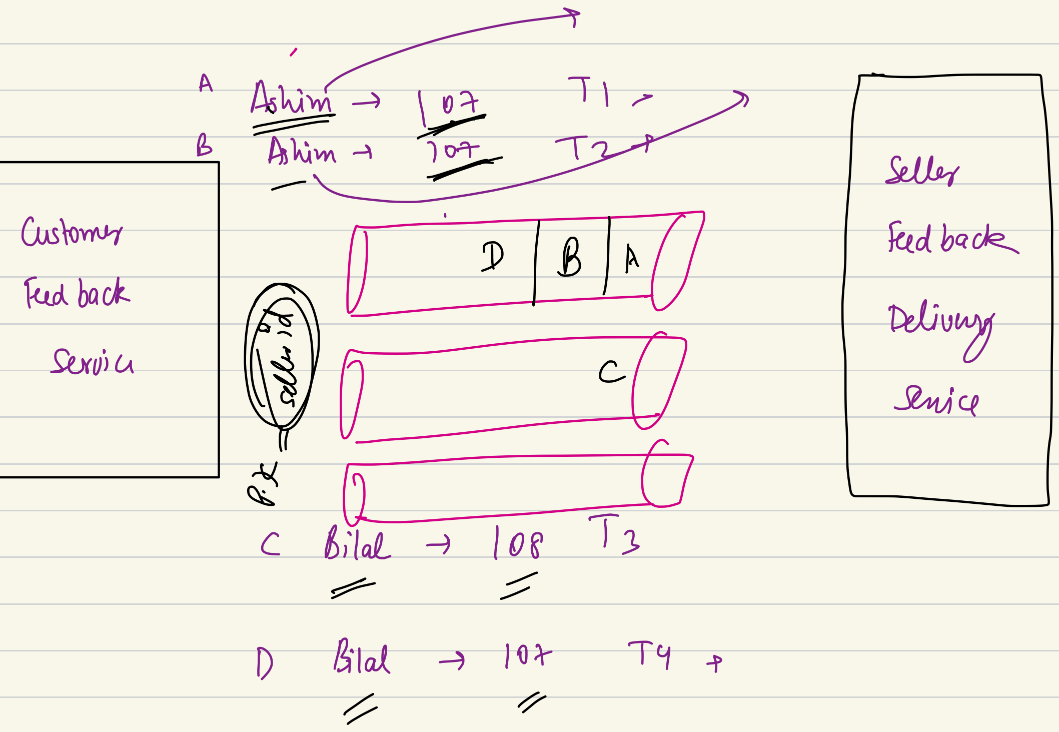
Only for msgs of the same partition, you can have FIFO gaurantees.

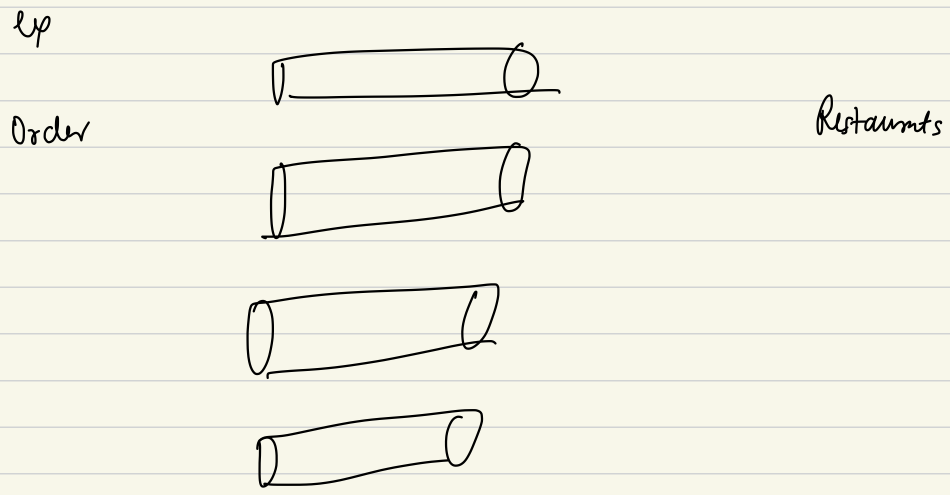


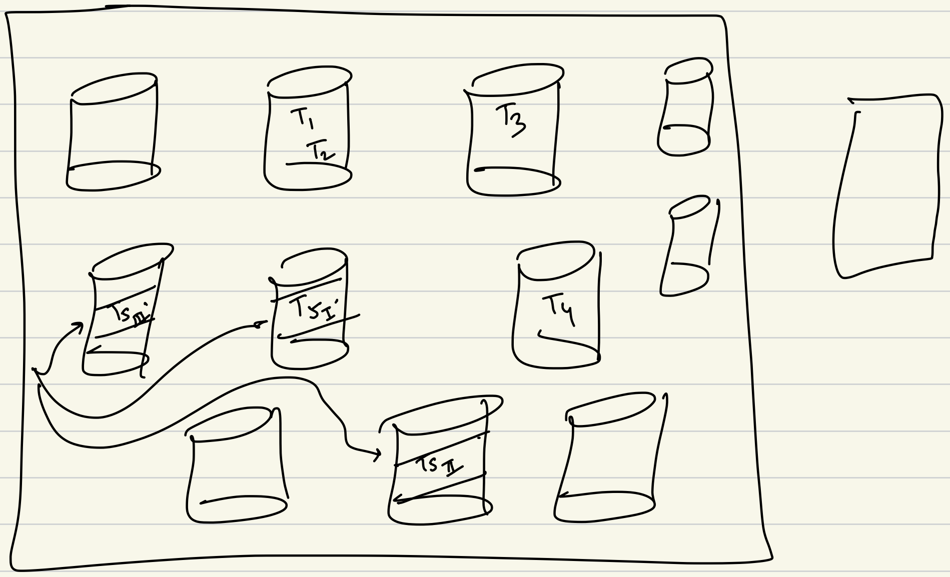


Order within the same partition is NOT lost;
but global ordering is LOST

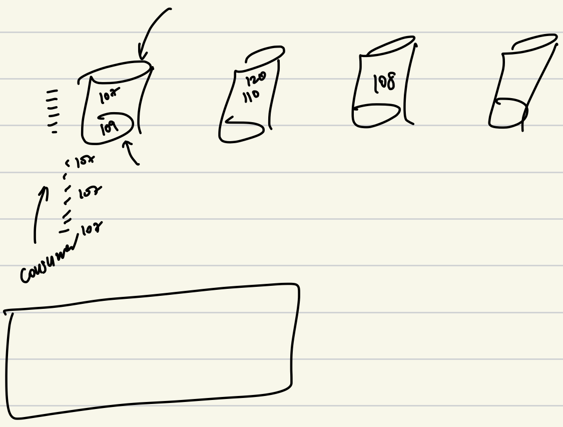
Kafka has to live with it


A Ashim → 107 T1
B Ashim → 107 T2
Customer
Feed back
Service
seller id
P.K
D
B
A
C
Seller
Feed back
Delivery
Service
C Bilal → 108 T3
D Bilal → 107 T4


ep
Order
Restaurants



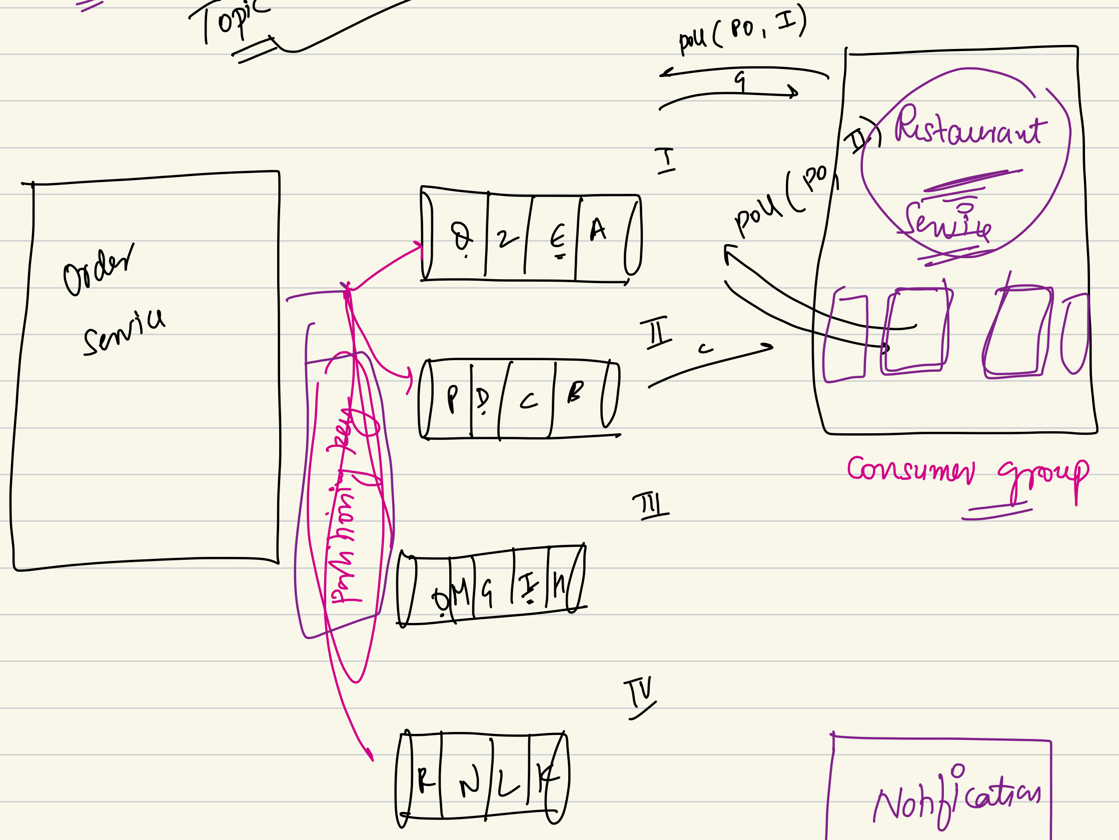
T1
T2
T3
T5 III
T5 I
T4
T5 II

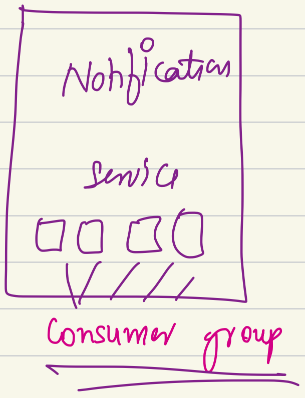


107
109
120
110
108
< 105
105
107
consumer

SWIGGY

Topic = Placed orders → 4 partitions





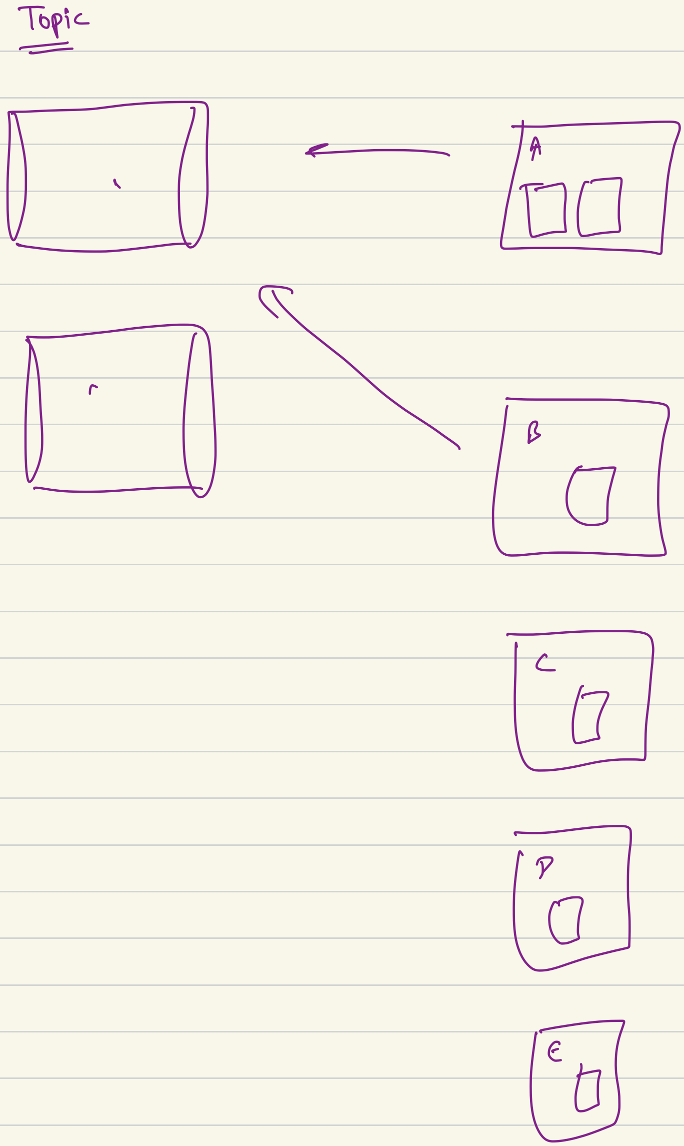
R.S

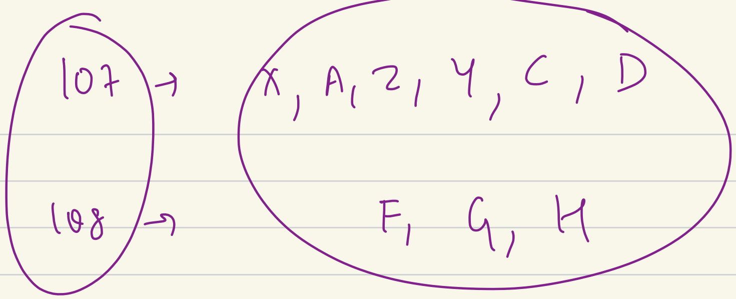
I	→	~~Q~~ 2
II	→	~~B~~ C
III	→	G
IV	→	N

NS

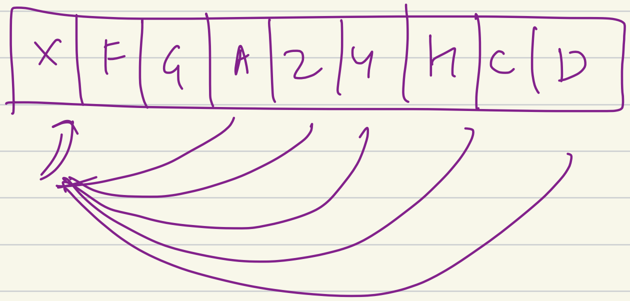
I	—
II	—
III	—
IV	—

Consume
5 services

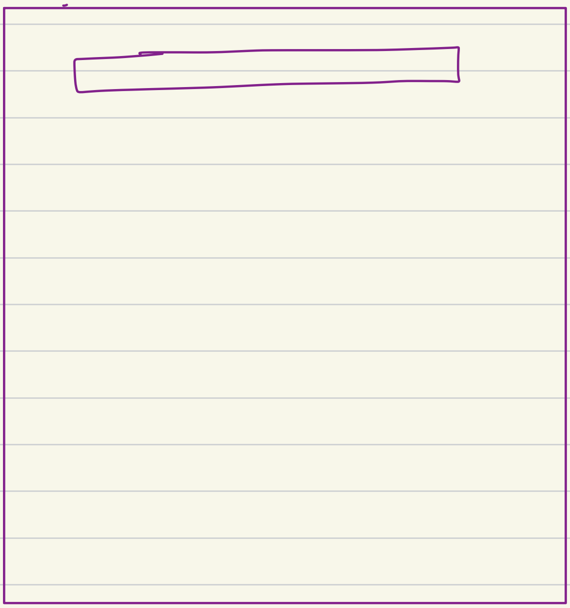




$P(107+108) \rightarrow$



Elastic Search



LinkedIn

$$\begin{tikzcd}
 & {} \arrow[dl] \arrow[dr] & \\
\text{Parametrized Searches} & & \text{Full Text Search}
\end{tikzcd}$$

A POST: India is a beautiful country

B POST : Canada India relationship is at a new low

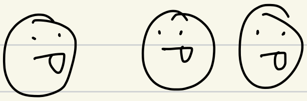
C POST : Canada Engineering colleges might be a sham.

D POST: India lost the CWC

E POST : I love my job. I love my life.

query: (India cricket)

SQL



select * from CONTENT_TABLE

where content LIKE % india%cricket%"

NoSQL

→ Document DB seems to be the most relevant DB out of all the DBs we studied to store above posts/web-pages

But still full text search is a challenge. it is unoptimized to perform it.

FIRST PRINCIPLES ⇒

$\boxed{\text{Inverse Index}}$

Glossary

Ram: 8, 10, 20, 24, 26

Aniket: 8, 80, 98, 101

Akash: 1, 100, 101

A POST: $\boxed{\text{India is a beautiful country}}$

B POST : Canada India relationship is at a new low

C POST : Canada Engineering colleges might be a sham.

D POST: India lost the CWC

E POST : I love my job. I love my life.

Inverted Index

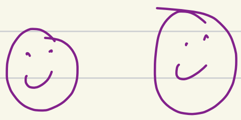
India	-	A, B, D
Canada	-	B, C
Country	-	A
Beautiful	-	A
relationship	-	A
Cricket	-	$\bigcirc$D
CWC	-	D

query:

"Cricket India"

Union + ranking

How to create an inverted index.



$\boxed{\text{Apache Lucene}}$

1 STOP WORD REMOVAL

a, an, the

2 STEMMING

→ run, running, ran

→ crash, crashed, crashes

→ USofAmerica, United States of America, USA

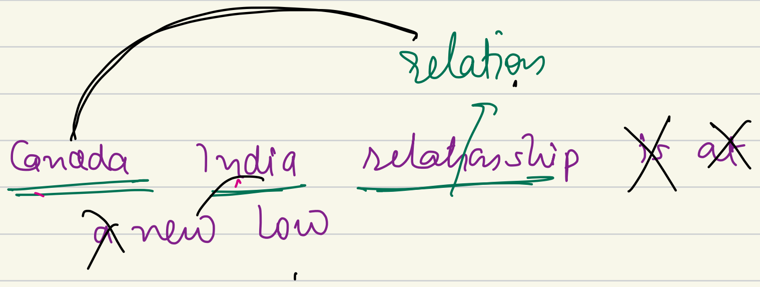
happy happiness happier → happy

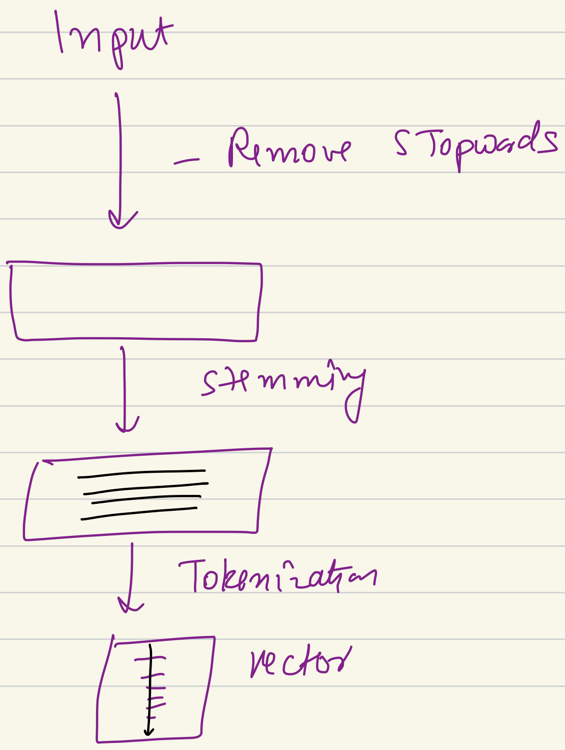
India, Indian, Indianess

3

③ Tokenization

POST B: Canada India relationship is at a new low





Unigram

Canada

India

relation

new

low

Bi-grams

Canada India

India relation

relation new

new low

Trigram

n-gram

5-gram

bi - skip - grams (1)

Tri - skip - grams (1)

Canada India relationship is at a new low

after Tokenization

[
Canada

India

relation

new

low

Canada India

India relation

relation new

new low
]

Content

↓

Vector of Tokens

Canada relation
India new
relation low

LinkedIn

1M posts

1 → Post ; Vector
2 → Post ; vector
3
4

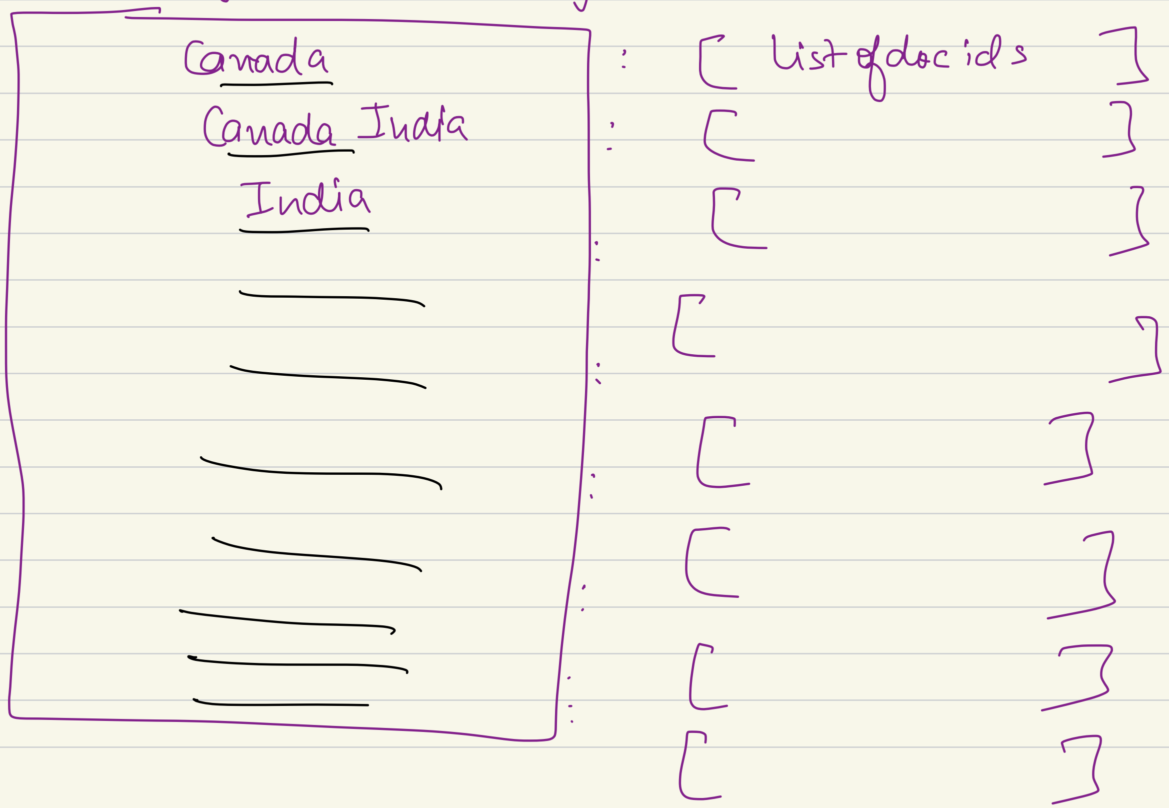
1M → post , vector

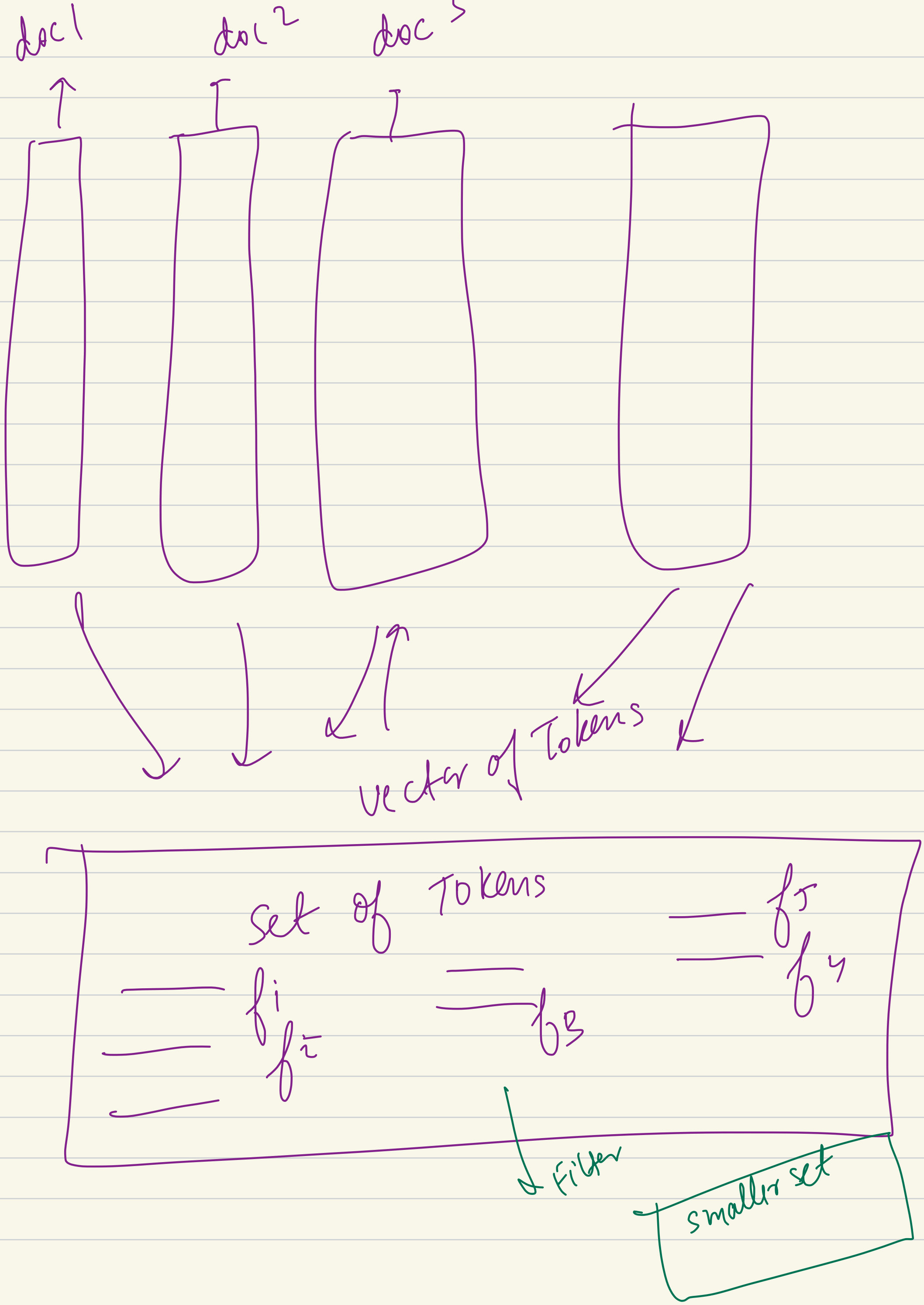
$$\boxed{\begin{array}{c}\text{All Tokens created globally} \\ \downarrow \\ \text{Assign some importance score} \\ \text{to each Token}\end{array}}$$

Tf - IDf

$$\begin{tikzcd}
 & \text{1M Documents} \arrow[dl] \arrow[dr] & \\
\text{raw document} & & \text{processed vector}
\end{tikzcd}$$

$$\begin{tikzcd}
\text{1M P. vectors} \arrow[d, "\text{Filtration}"] \\
\text{a few million interesting Tokens}
\end{tikzcd}$$




doc 1
doc 2
doc 3
vector of Tokens
set of Tokens
f1
f2
f3
f5
f4
Filter
smaller set