data -> processing -> result
2 records -> spark -> res1 [hostorical, dbase, datalake, nosql ] -> streaming
3 records -> spark -> res2
10 records -> spark -> res

T1
empid, sal,empName
1,100
2,200

3,300
4,400
5,500

T2
empid, dept
1,d1
2,d2

3,d3
4,d4
5,d5

select empName,dept from T1 right join T2 on T1.empid = T2.empId

10 records -> spark -> storing this data on hdfs
-> doing lookup in database and getting some recommendation -> storing this recommnded data in nosql

-> sending back this recommendations to the user (restApi)

credit card ->

Distributed
fault T
HA
zookeeper

producer -> KAFKA <-
consume(up n running)
(p4,p3,p2,p1)

Broker M1
(Topics

consumer grp = [c1(333), c2(333),
c3(334), c4] - point to point
communication [MSQ]

p1 - Topic1(internet banking) (queue) <-
c1(1000). public sub model
<-c2
(1000)
Partitioning

3 - part0, part1, part2
key = hash mechanish
hashcode % 3 =
round robin -
message 1 -> part0
message 2 -> part1

messgae 3 -> part2

message -> part0

p1 = producer(“Topic1”, message1, “k1”)
p1 = producer(“Topic1”, message1, “0”)

p2 - Topic2(credit card) <-c2
p3 - Topic3(debit card) <-c3

p1 = producer(“Topic1”, message1)
p1.send()

c1 = cosumer(“Topic1”)

p2 = producer(“Topic2”, message2)
p2.send()
c2 = cosumer(“Topic2”)

)