



CS 165

Data Systems

Have fun learning to design and build modern data systems

class 12

scans vs indexes

prof. Stratos Idreos

[HTTP://DASLAB.SEAS.HARVARD.EDU/CLASSES/CS165/](http://DASLAB.SEAS.HARVARD.EDU/CLASSES/CS165/)



midterm reviewing next week

project testing

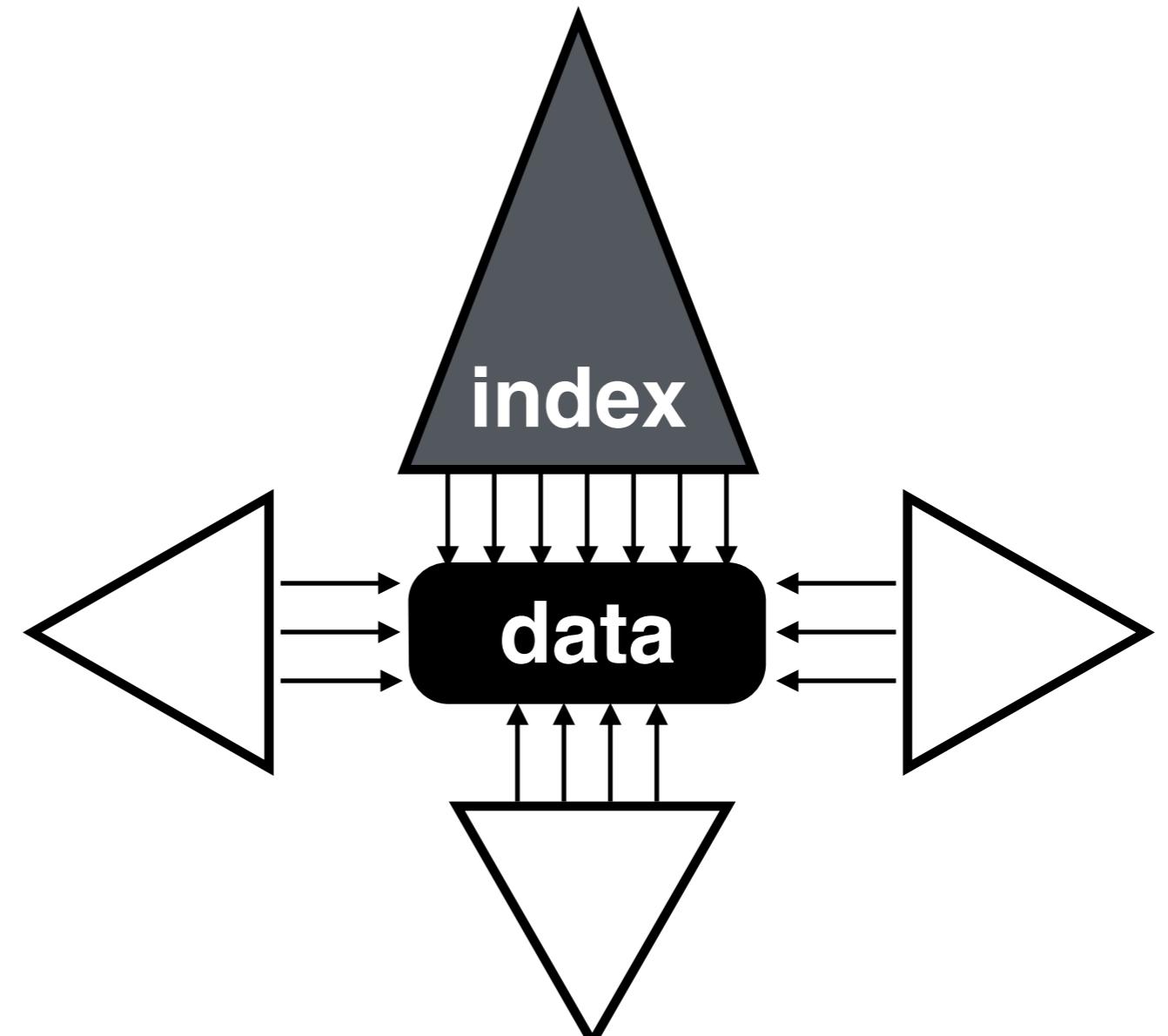
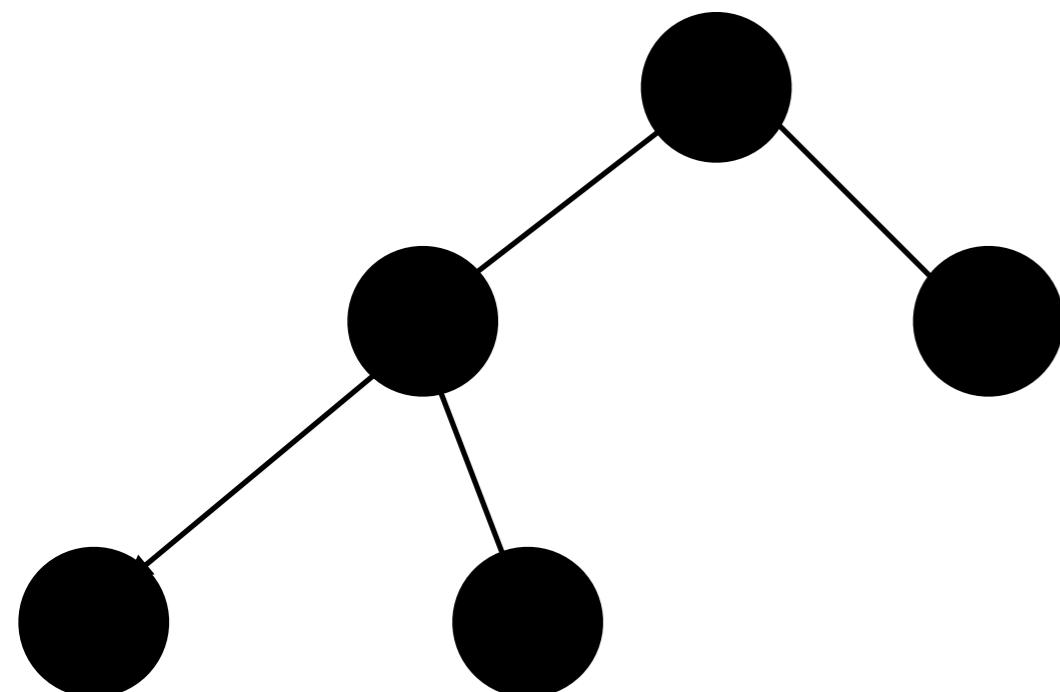
automated testing & leaderboard biweekly

your tests

SQL tests as of M3

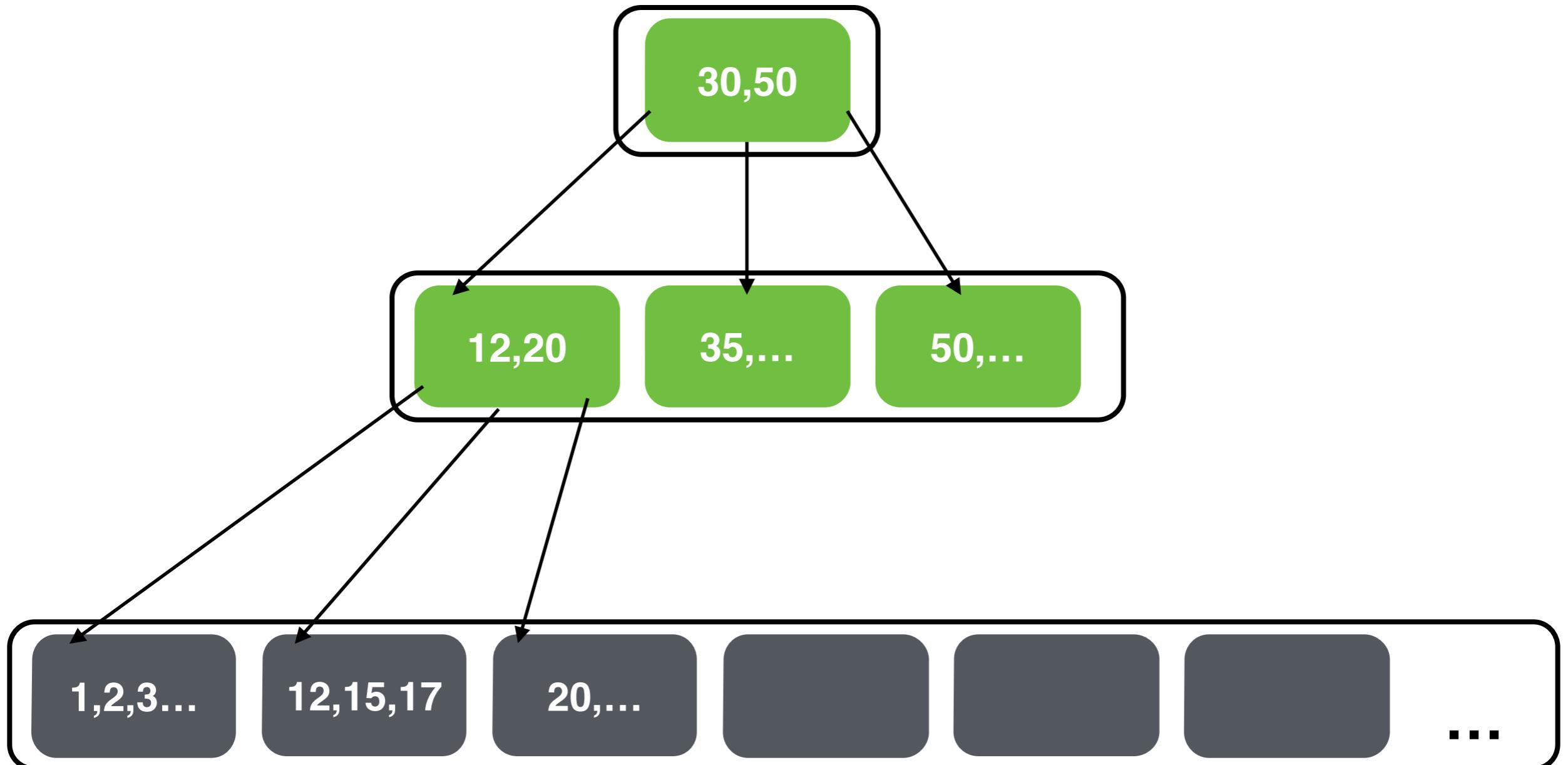
work with the TFs to stabilize testing procedures

clustered (all columns)



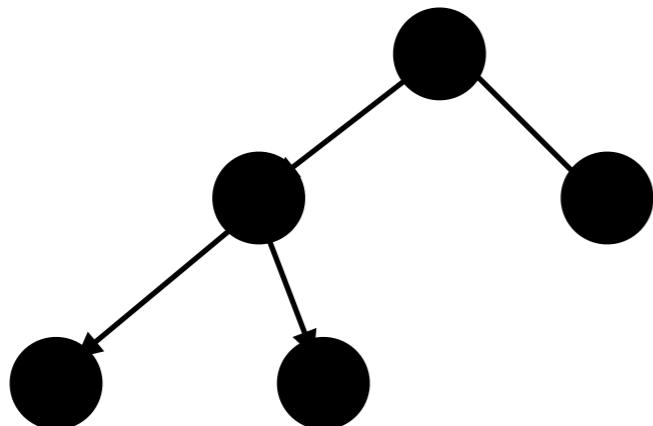
secondary indexes
subset of columns

b-tree - dynamic tree - always balanced



milestone 2

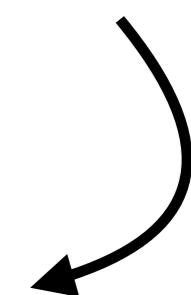
cache conscious b-tree-like index



node design,
fill factor, etc

most differences with classic design
you end up doing for the leaves

contiguous vs not, tuples vs single values, fill factor,
need to maintain alignment info with the rest of the columns



insert & select now, updates later

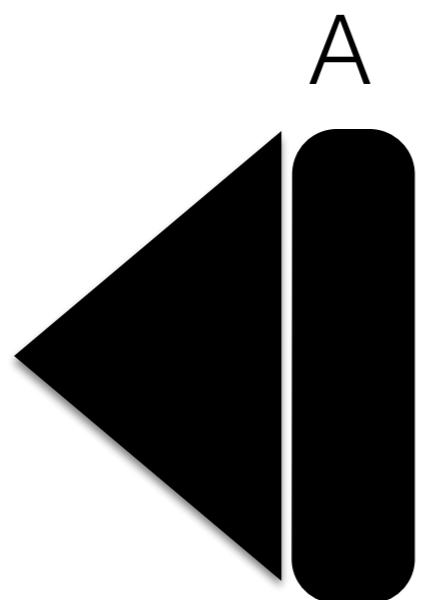
declarative interface
ask what you want

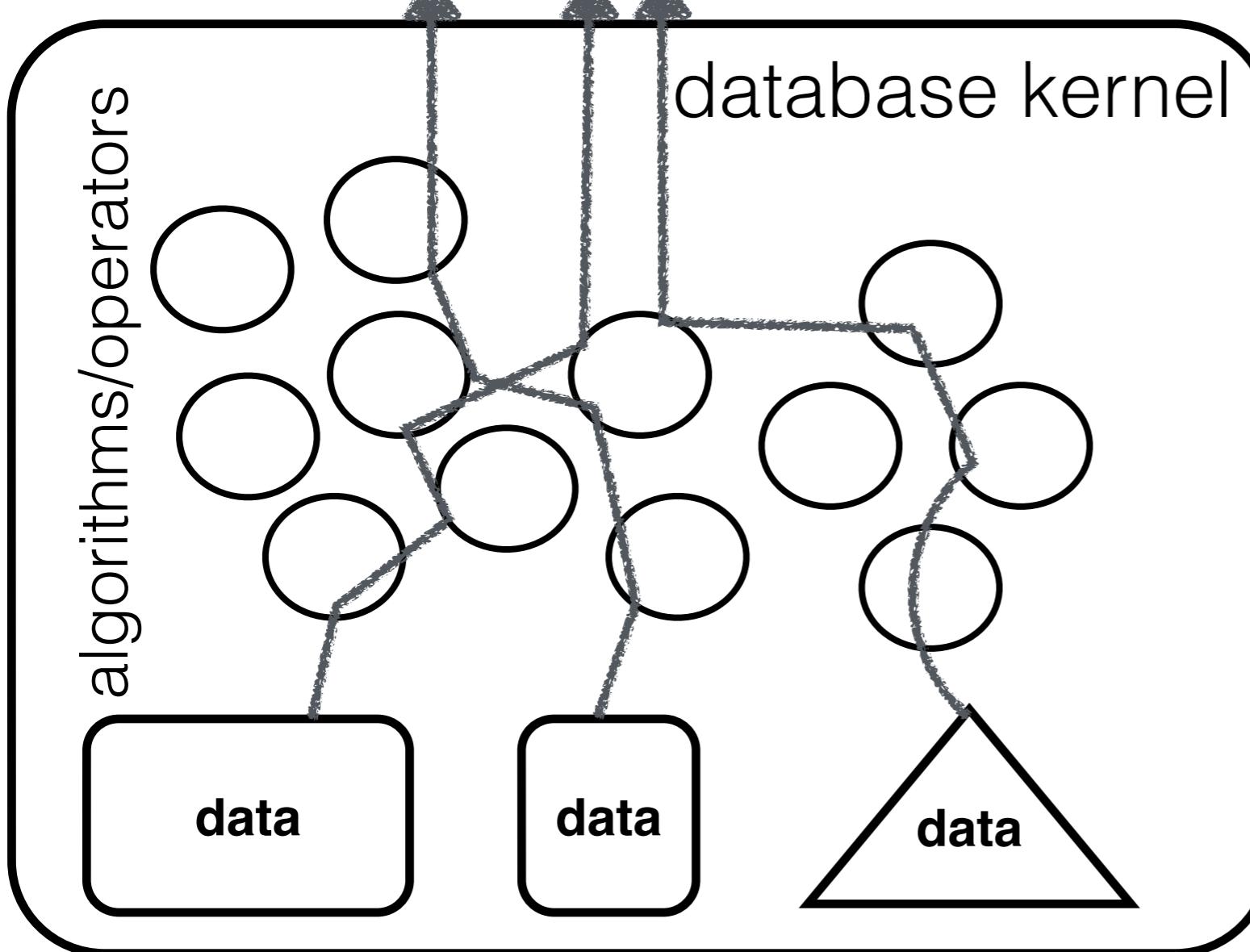
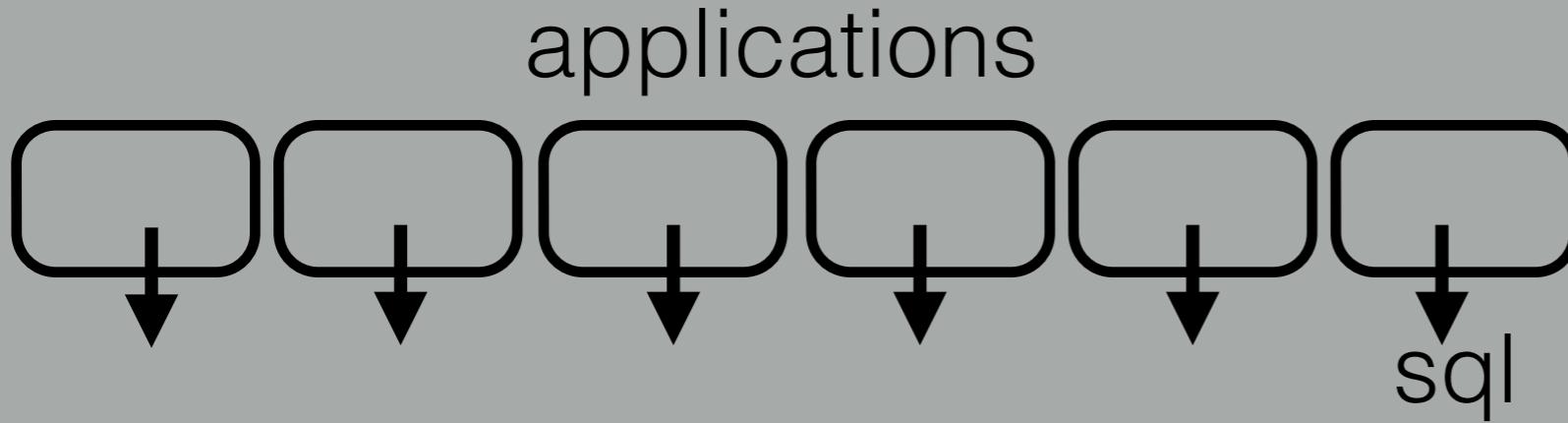


indexes/views/tuning knobs

select ... from R where A<v and

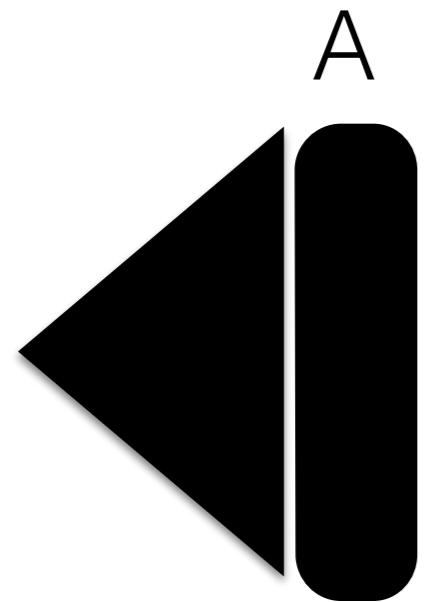
(secondary) **index vs scan**: the eternal battle





**design/implement
numerous**
possible algorithms
+ data representations

choose the best
data source, algorithms
and path for each query



random access
to traverse the tree
& need to sort result

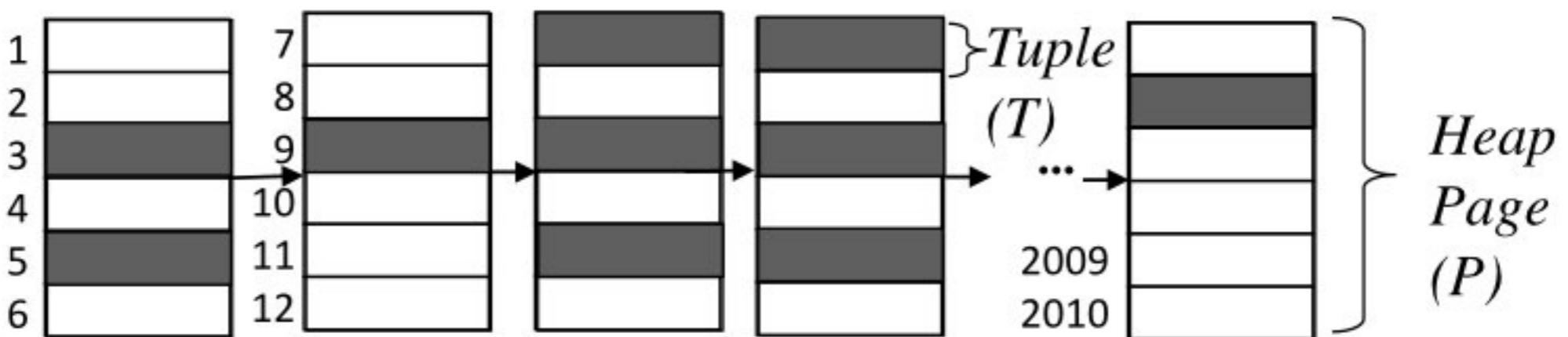


sequential access
pattern but needs to
access all data

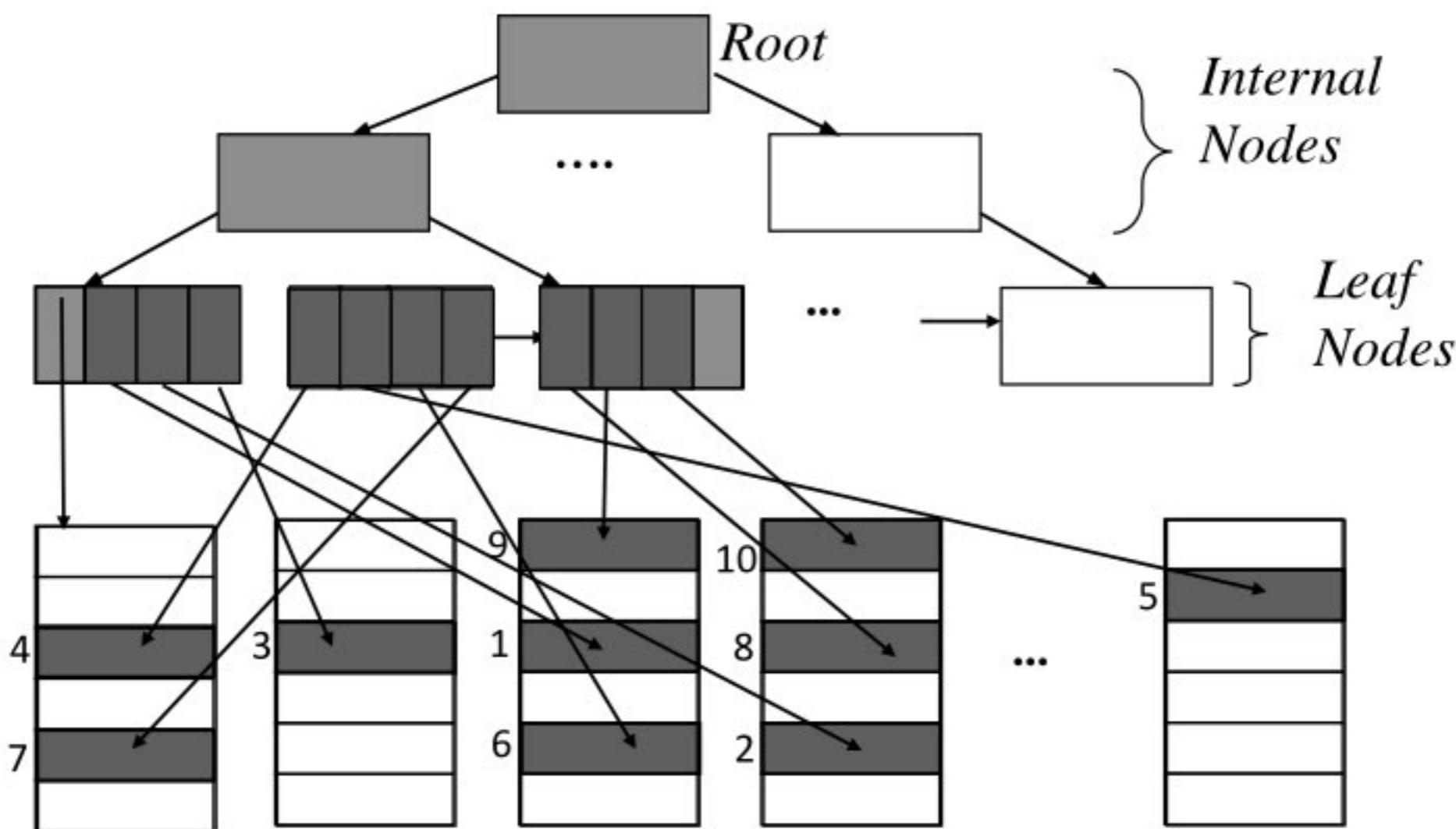


Table

scan



index
scan



A
a1
a2
a3
a4
a5

B
b1
b2
b3
b4
b5

C
c1
c2
c3
c4
c5

A
a5
a3
a2
a1
a4

secondary index on A
values out of order
with base data

a query that select on A and then needs B

intermediate out of order



A
a1
a2
a3
a4
a5

B
b1
b2
b3
b4
b5

C
c1
c2
c3
c4
c5

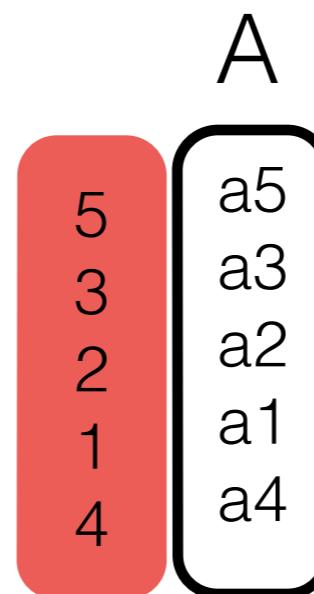
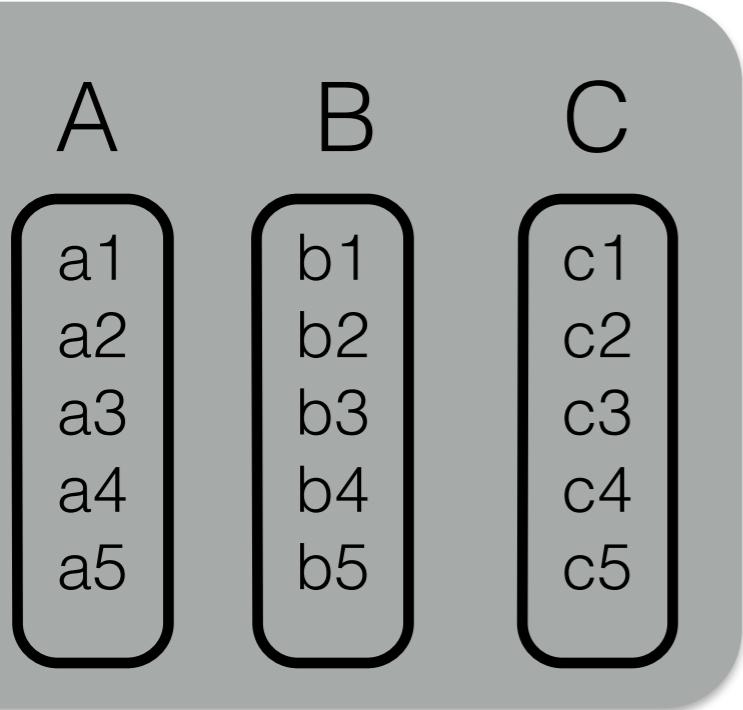
A
5
3
2
1
4

secondary index on A
values out of order
with base data

a query that select on A and then needs B

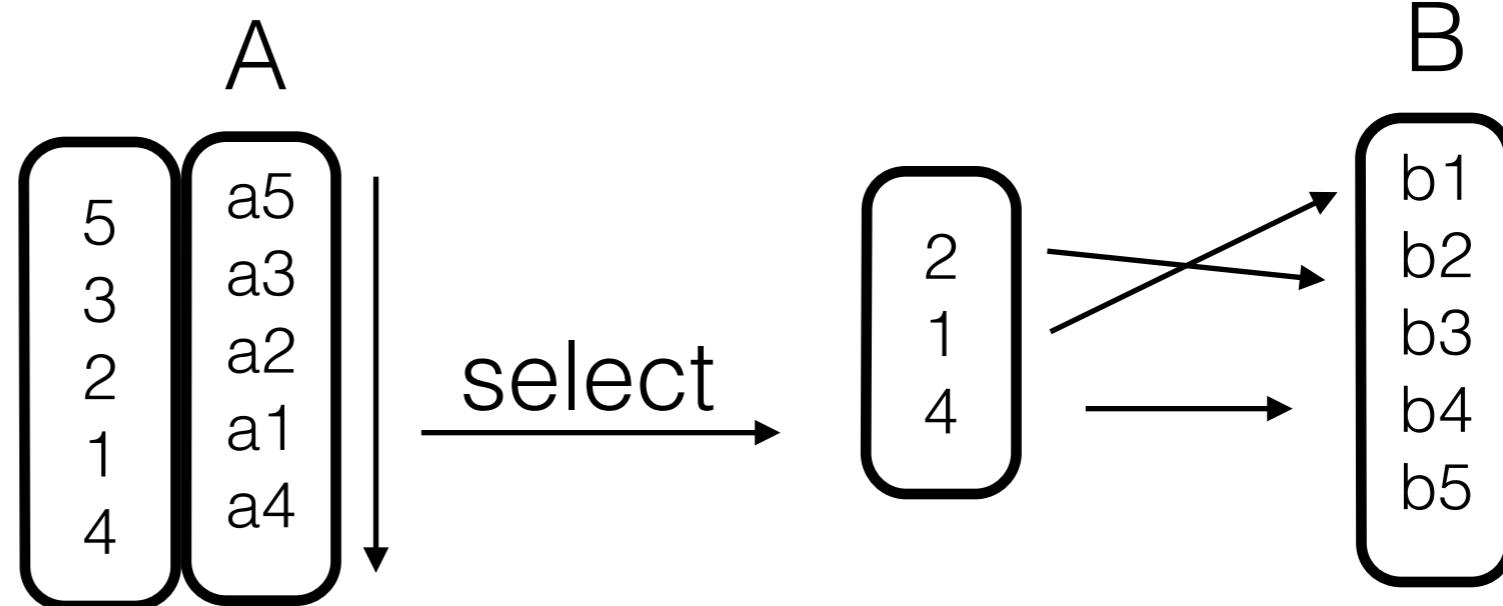
intermediate out of order





secondary index on A
values out of order
with base data

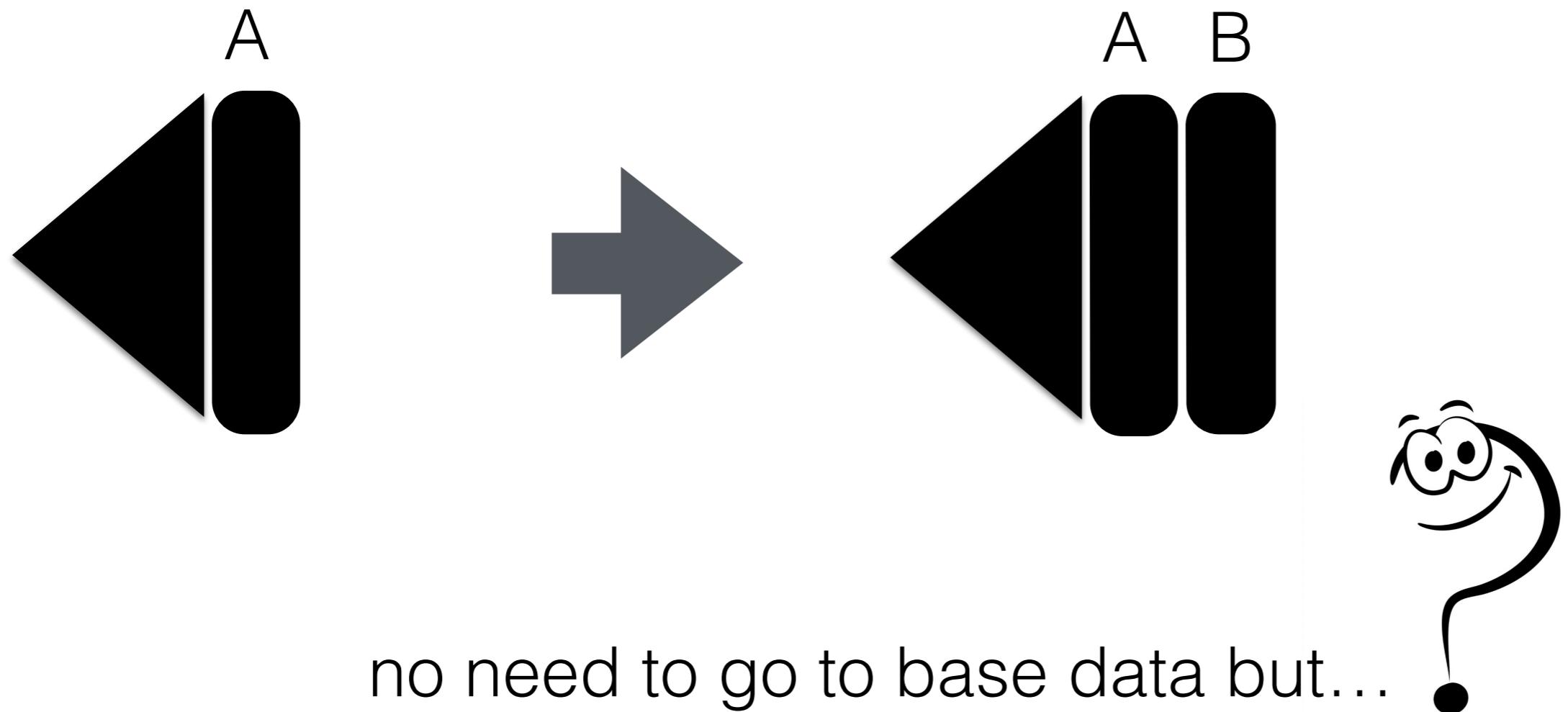
a query that select on A and then needs B

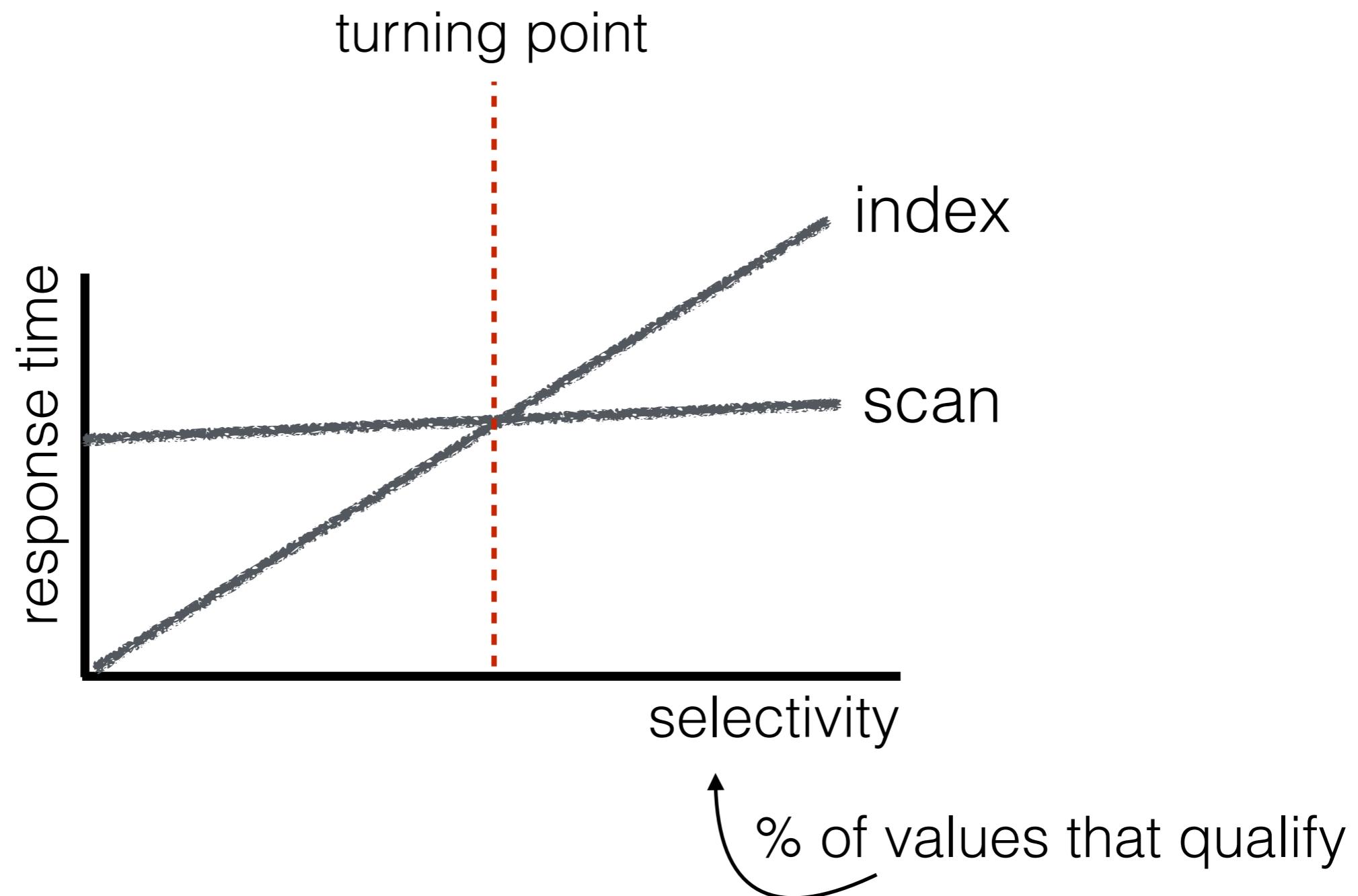


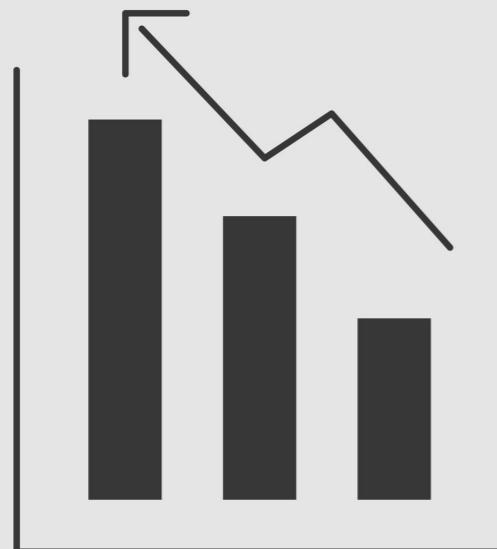
intermediate out of order



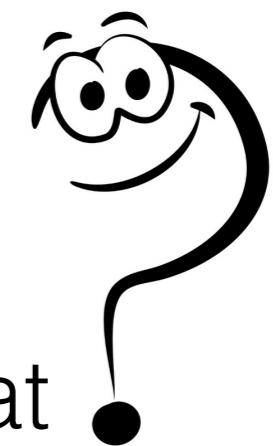
covering indices







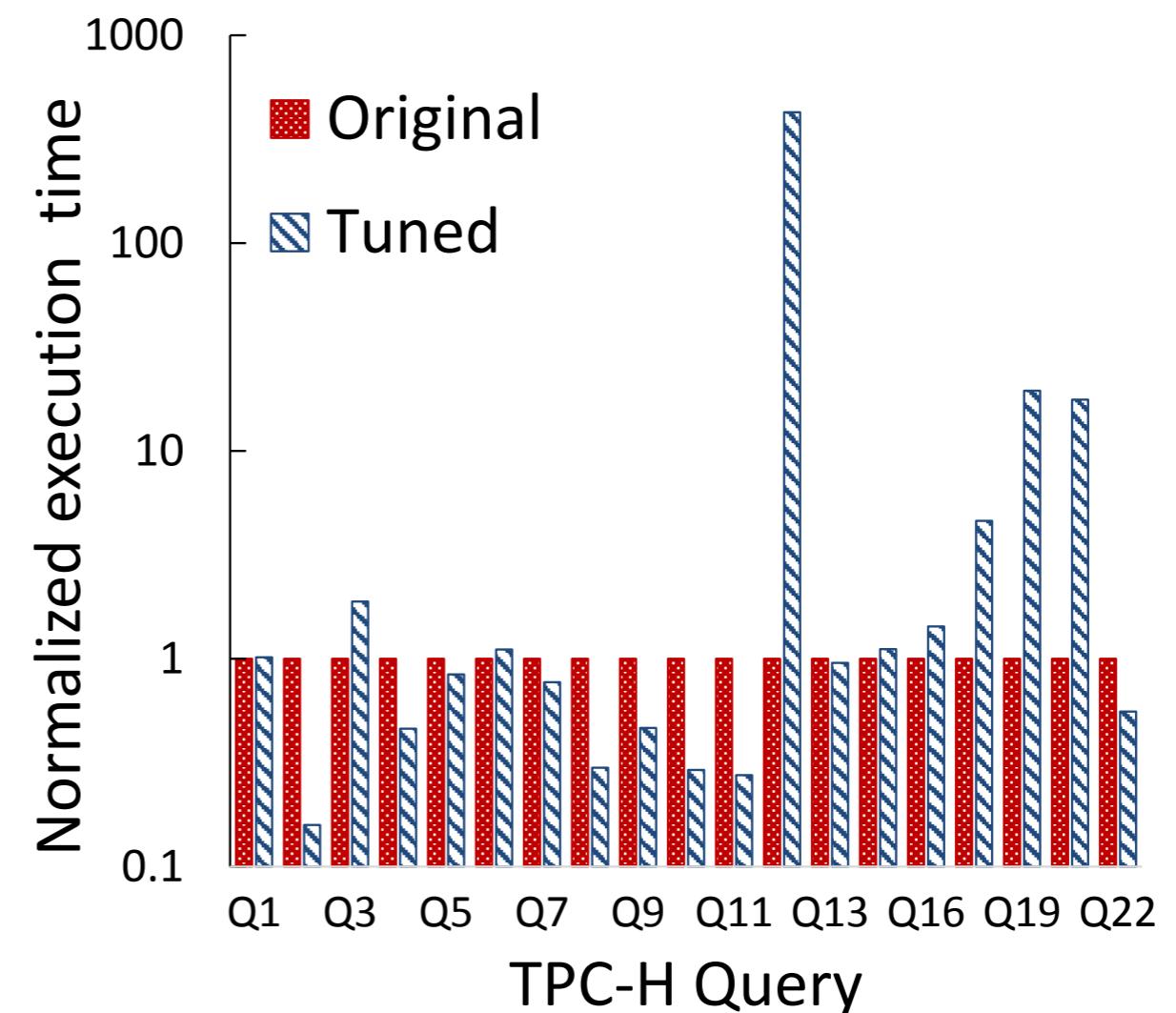
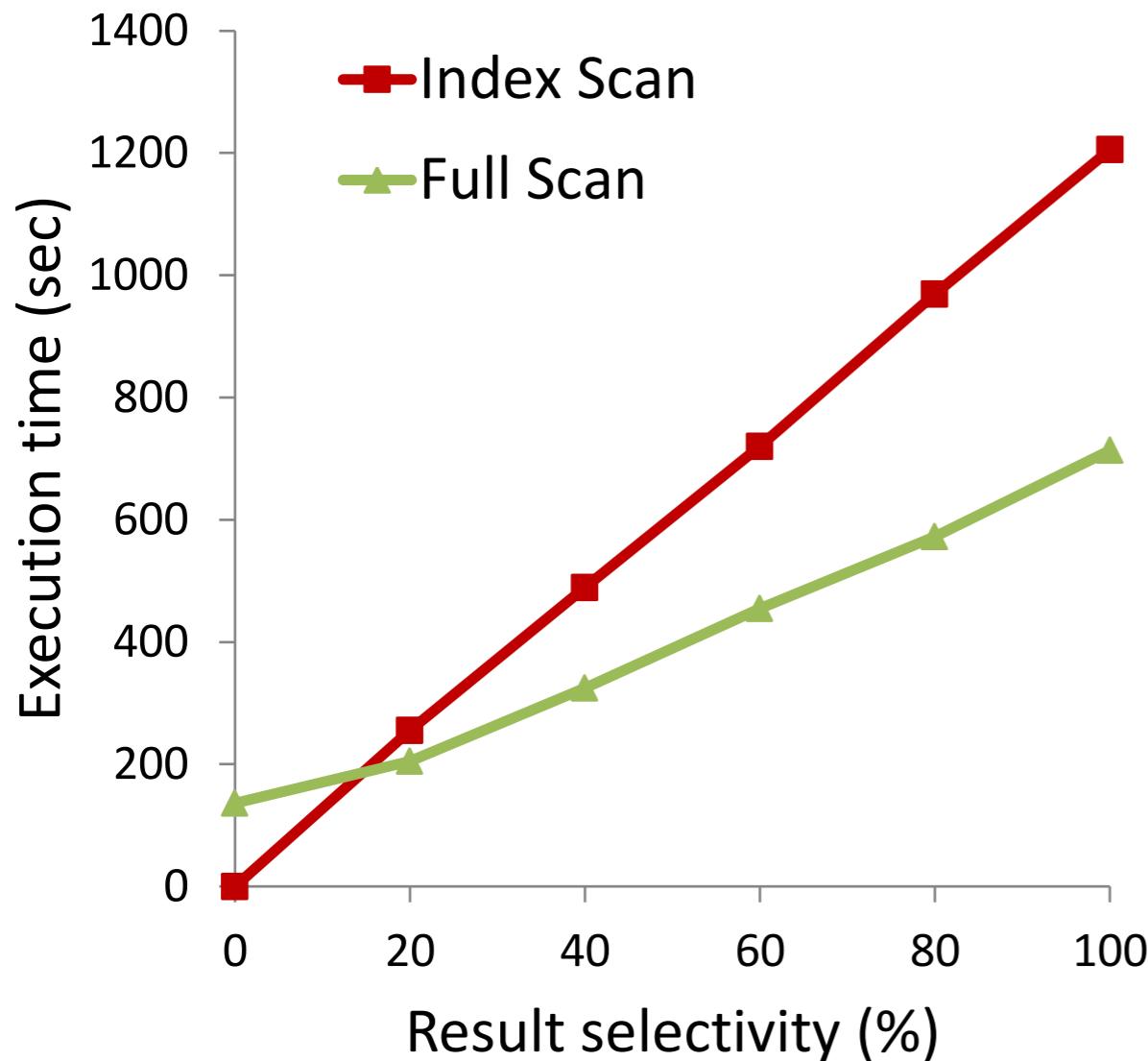
what is wrong with that



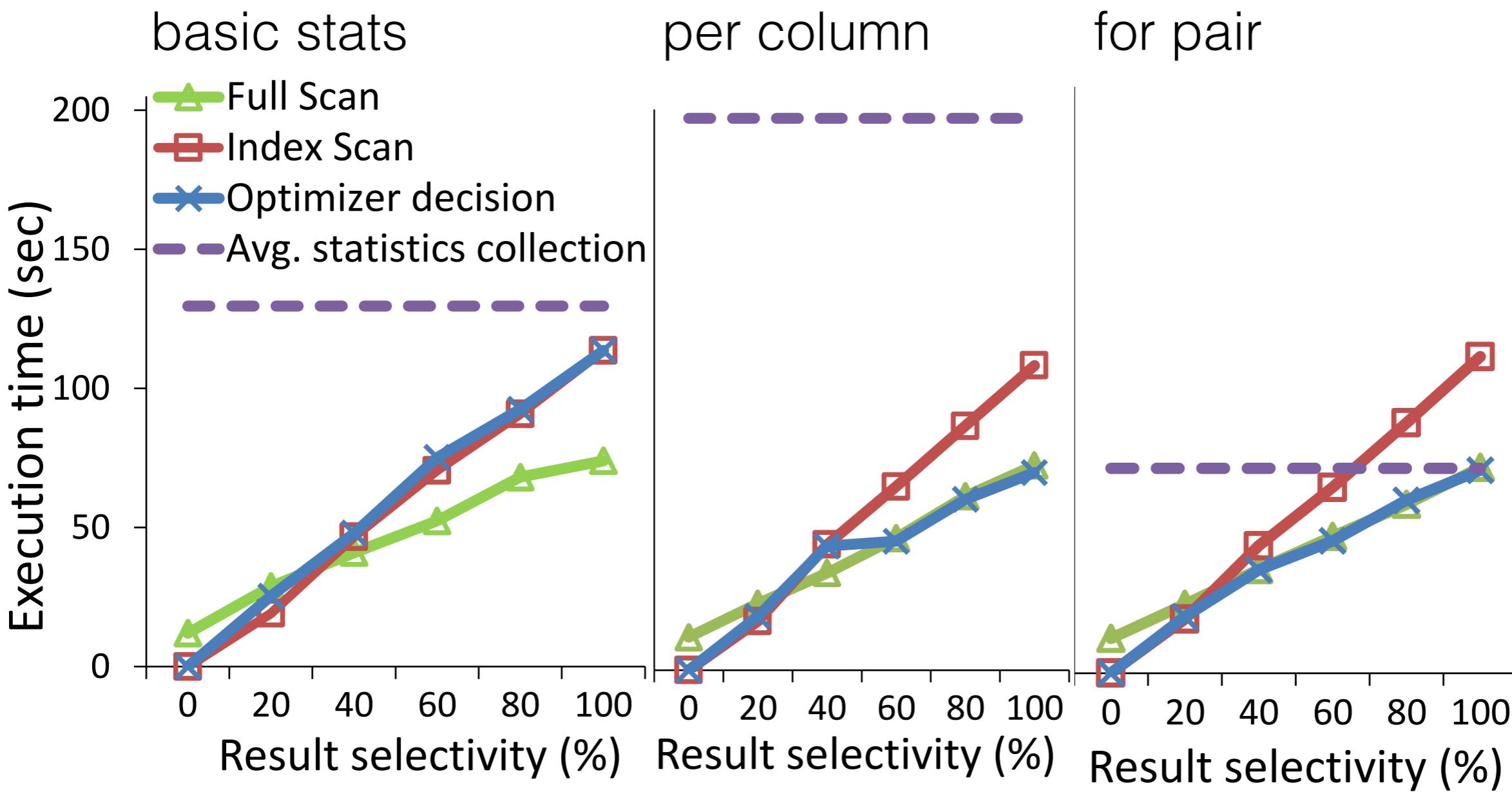
the standard solution

- 1) maintain statistics,
- 2) optimizer chooses access path depending on estimated selectivity





can we just recompute the statistics?





2012, somewhere in Germany



if I keep 30 data systems
researchers “trapped” in a castle
for a week we might be able to
define “robust query processing”
and find a few solutions :)





robust query processing (best definition to date by Goetz)
graceful degradation when the environment changes





Renata Borovica
EPFL



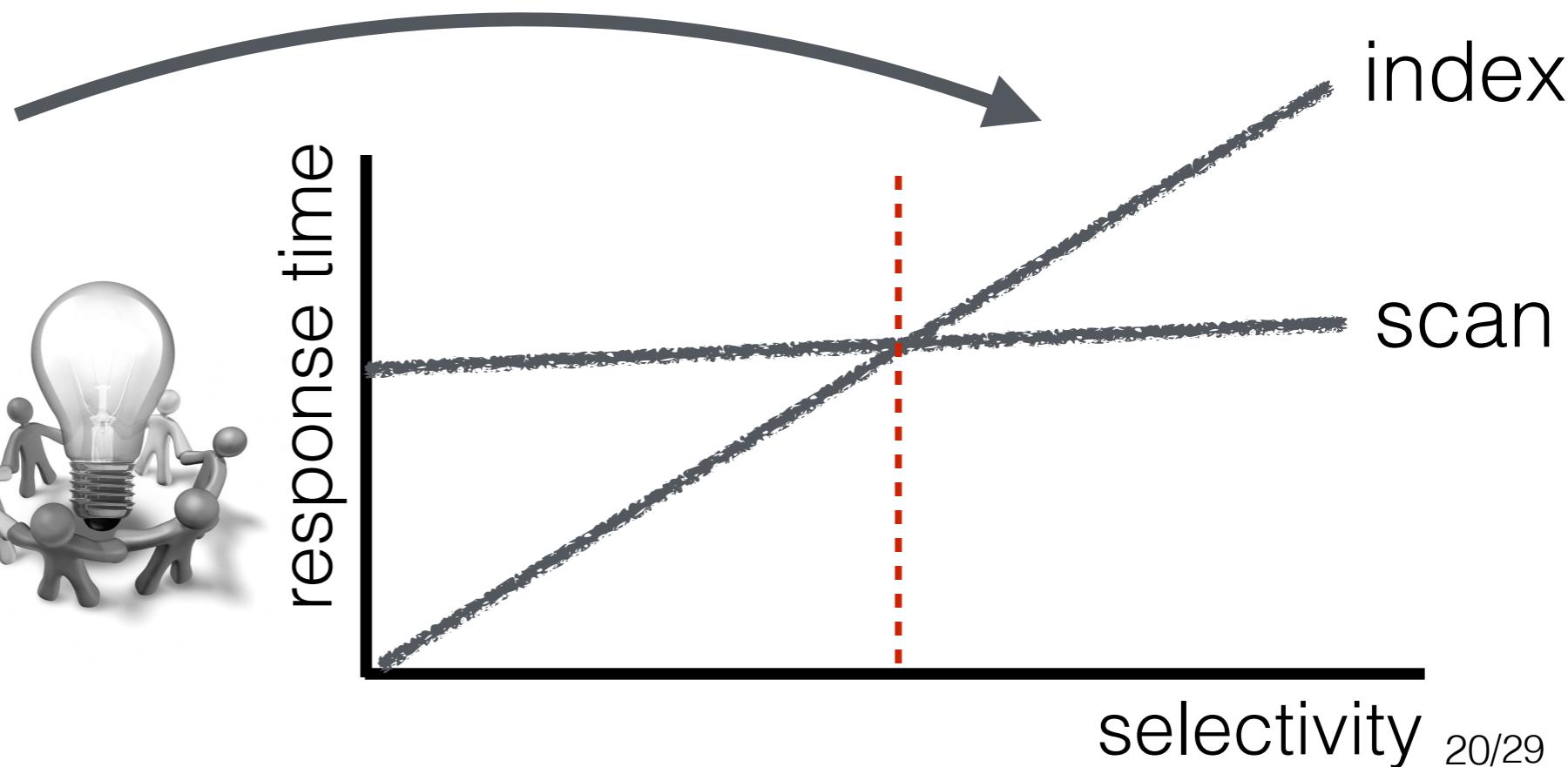
you should know
Marcin by now
Snowflake

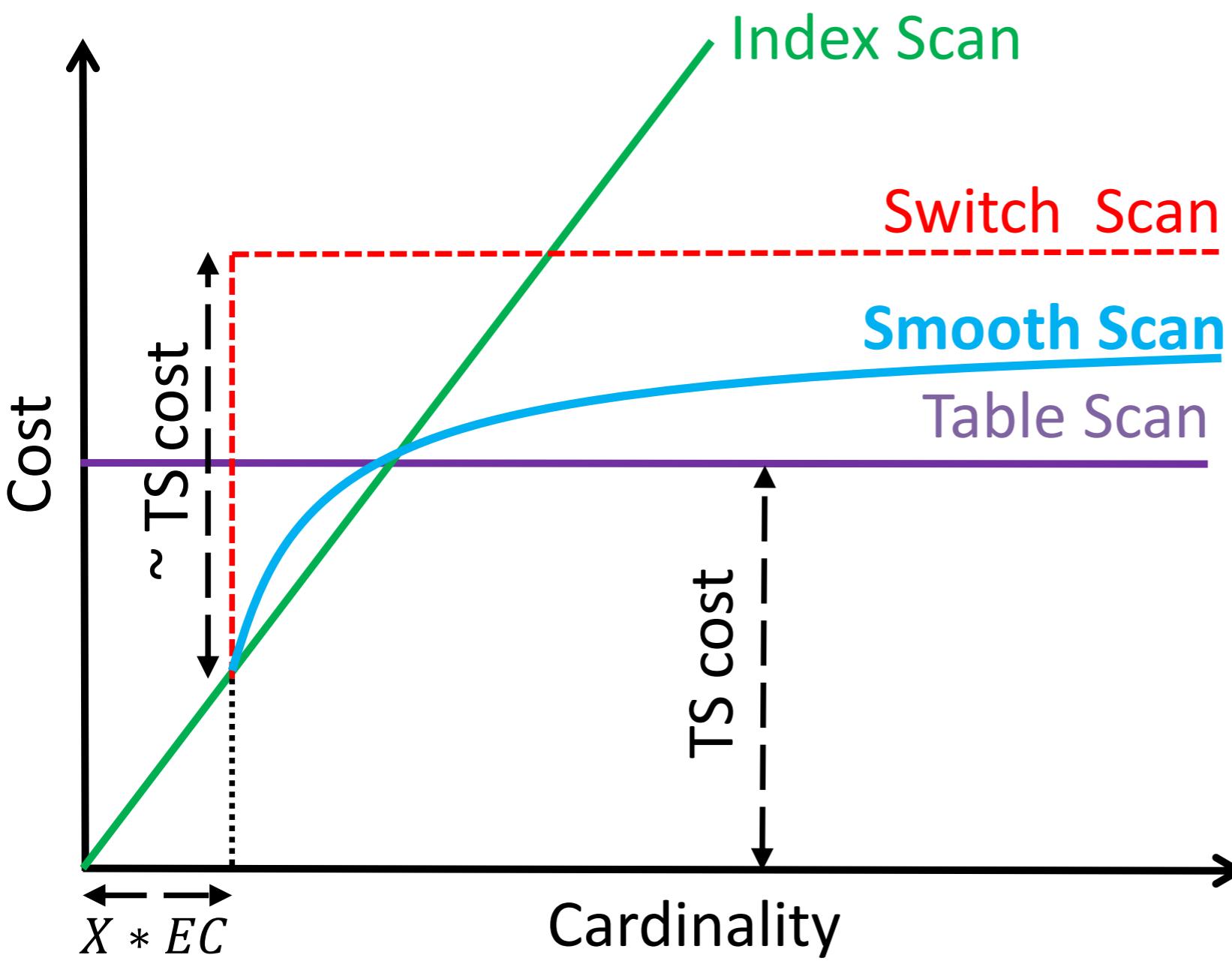


Campbell Fraser
Microsoft
now Google



**can we avoid bad
access path selection
even when we have
wrong statistics?**





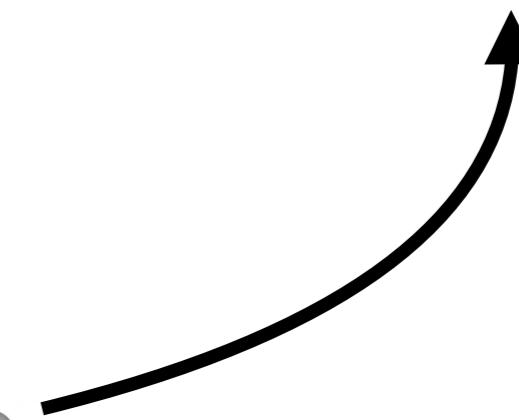
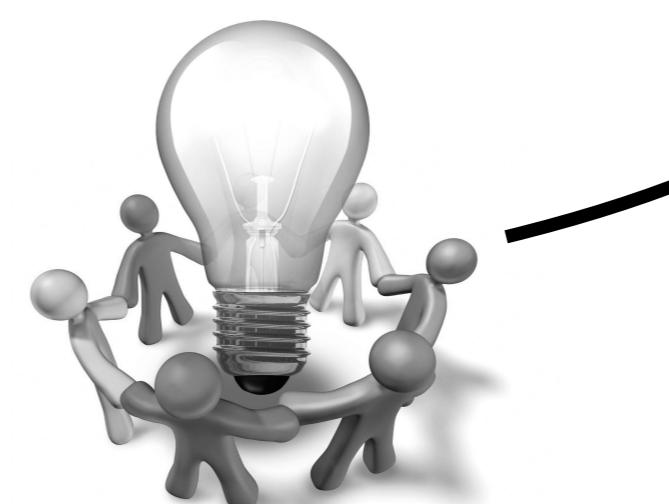
SWITCH SCAN

while index probing
switch to scan
if cardinality > estimation

good: avoids worst case
bad: performance cliff

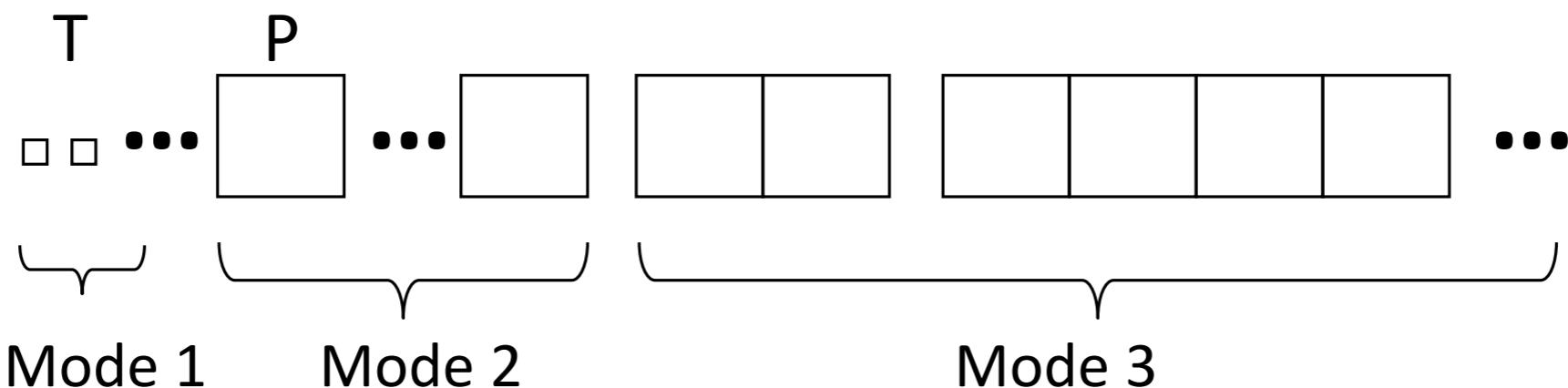
SMOOTH SCAN

goal avoid performance cliff
close to optimal



smooth scan

gradually morph from index scan to full scan



for each qualifying tupleID

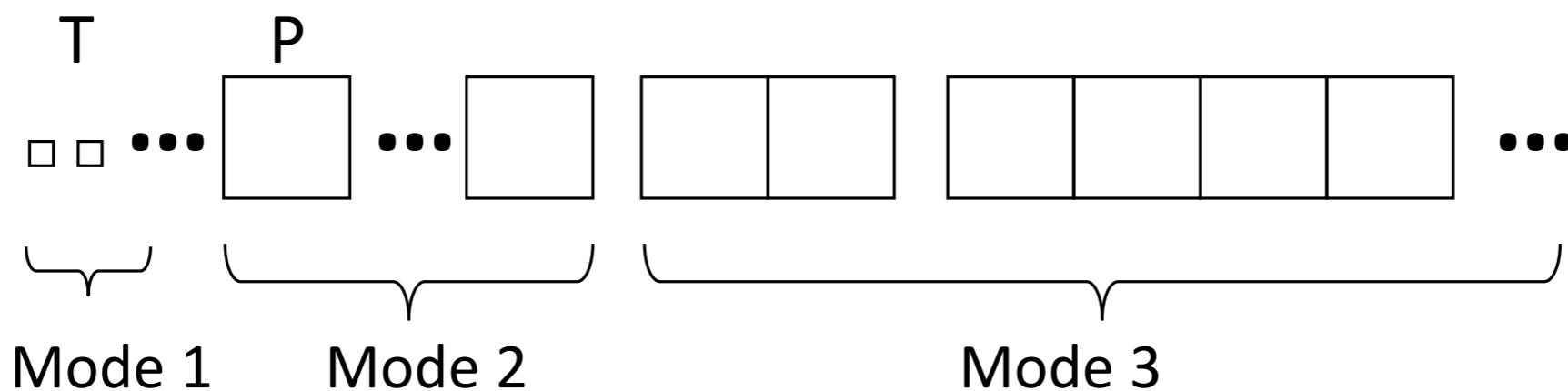
mode1 fetch the respective page and get the row

mode2 check all tuples in a fetched page

mode3 fetch and check adjacent pages as well

mode3+ increase # of pages fetched

some design points



tuple cache to avoid producing the same tuple twice
page cache to avoid reading the same page twice
result cache to produce result in indexed order

when to morph

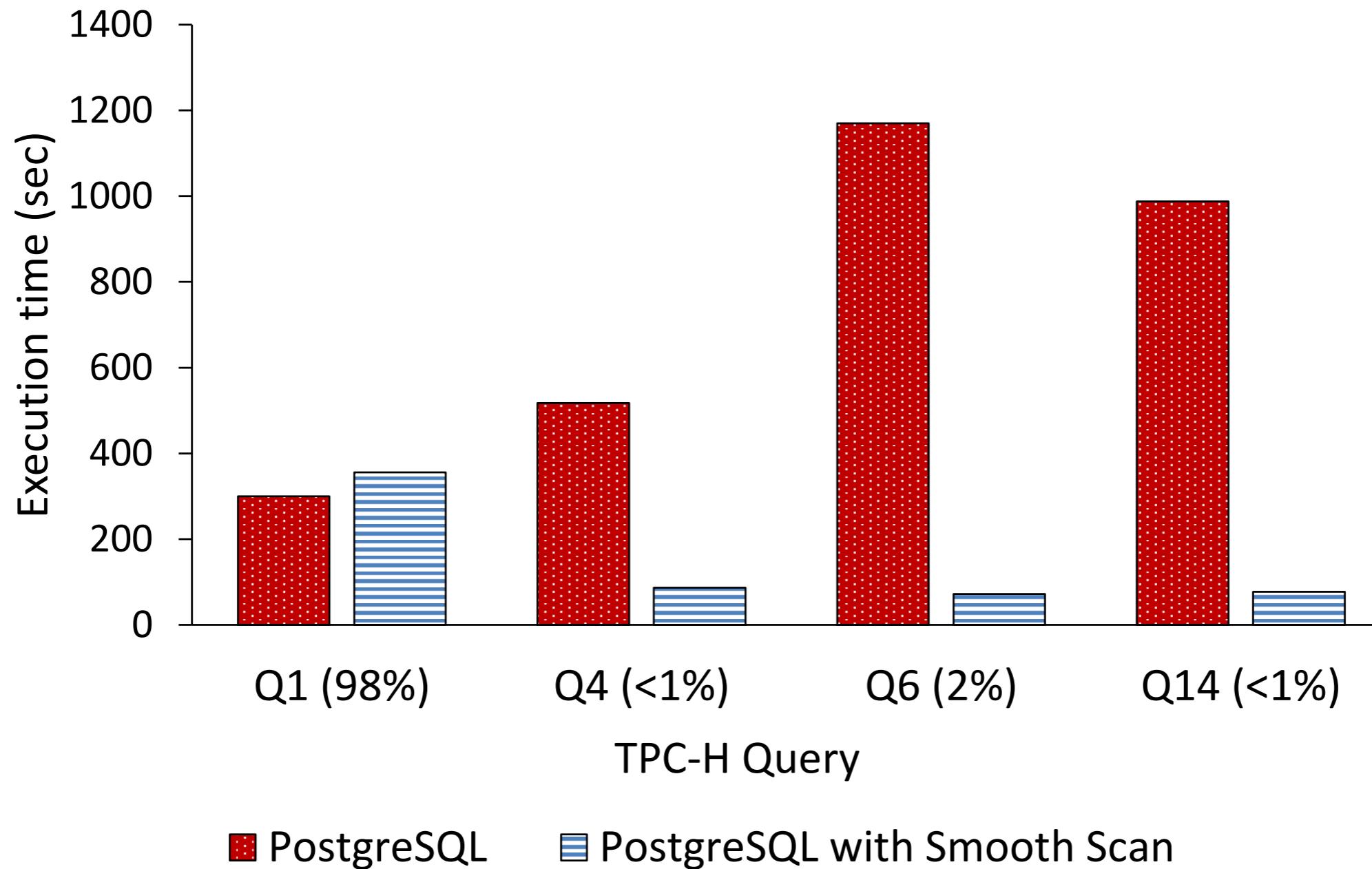
in order to achieve a smooth behavior

optimizer start when selectivity > estimation

SLA respect an upper threshold

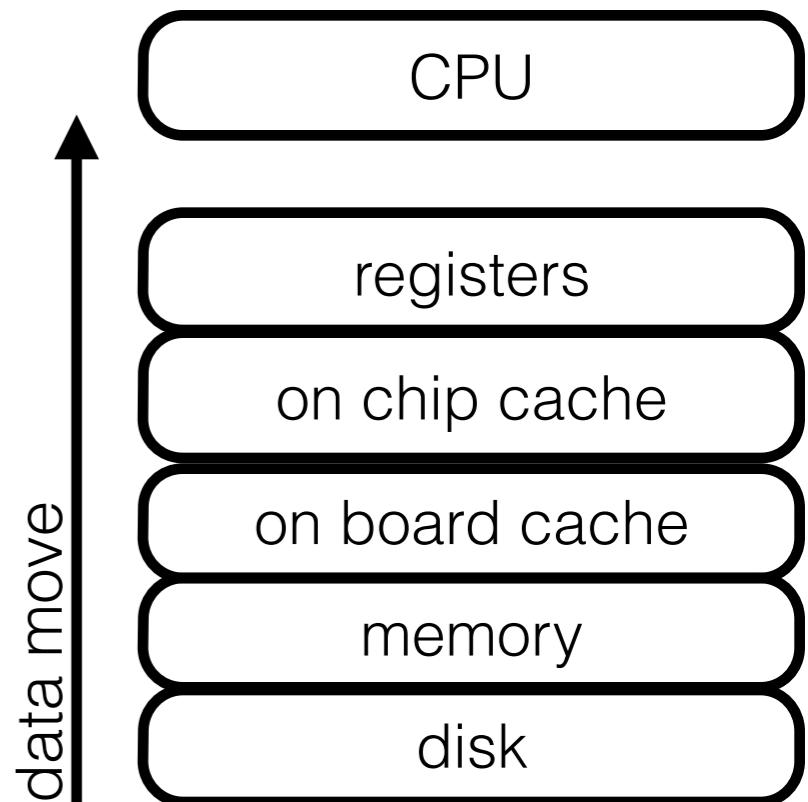
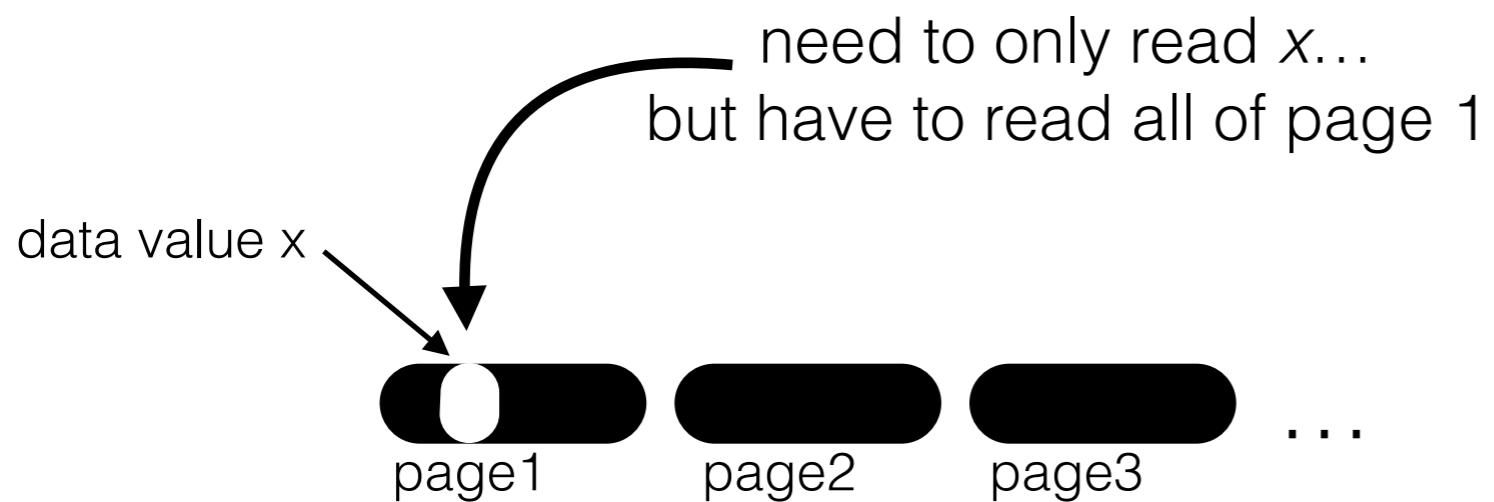
selectivity morph when selectivity increases by z

pessimistic morph with every new probe

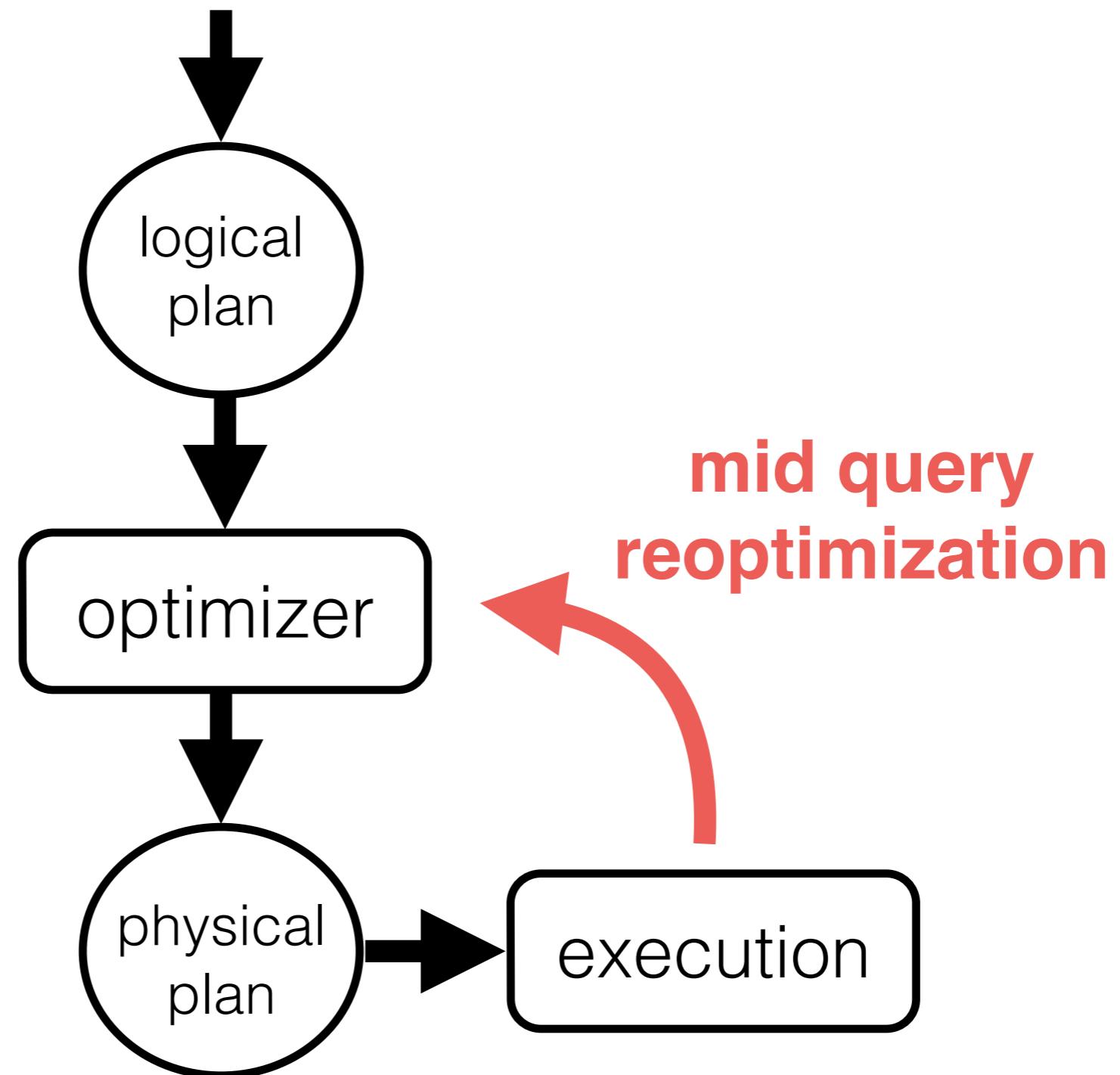




random access & page-based access



select min(A) from R where B<10 and C<80





Efficient mid-query re-optimization of sub-optimal query execution plans

Navin Kabra and David DeWitt

ACM **SIGMOD** International Conference on Management of Data, 1998

Smooth Scan: Statistics-Oblivious Access Paths

Renata Borovica, Stratos Idreos, Anastasia Ailamaki, Marcin Zukowski and Campbell Fraser
IEEE International Conference on Data Engineering (**ICDE**), 2015

next: fast scans

scans vs indexes

DATA SYSTEMS

prof. Stratos Idreos



HARVARD
School of Engineering
and Applied Sciences