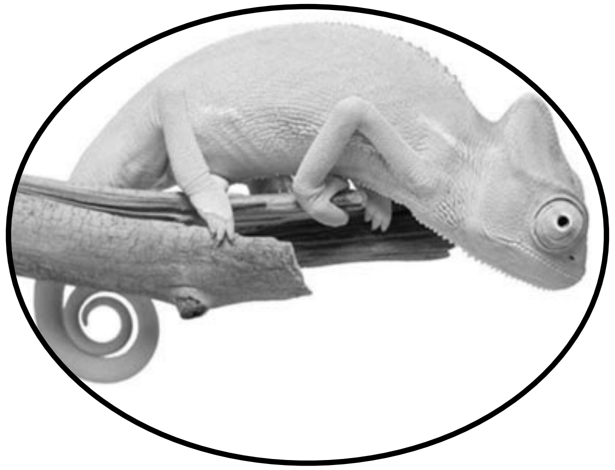
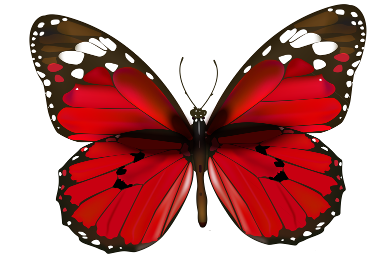
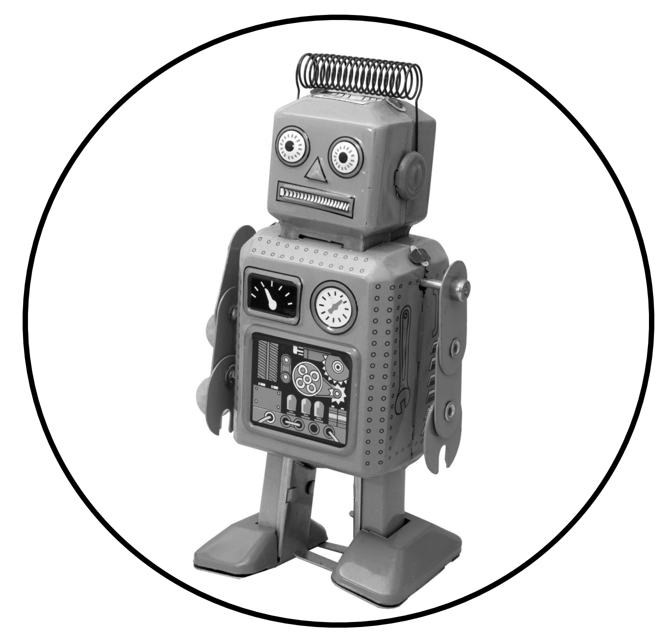




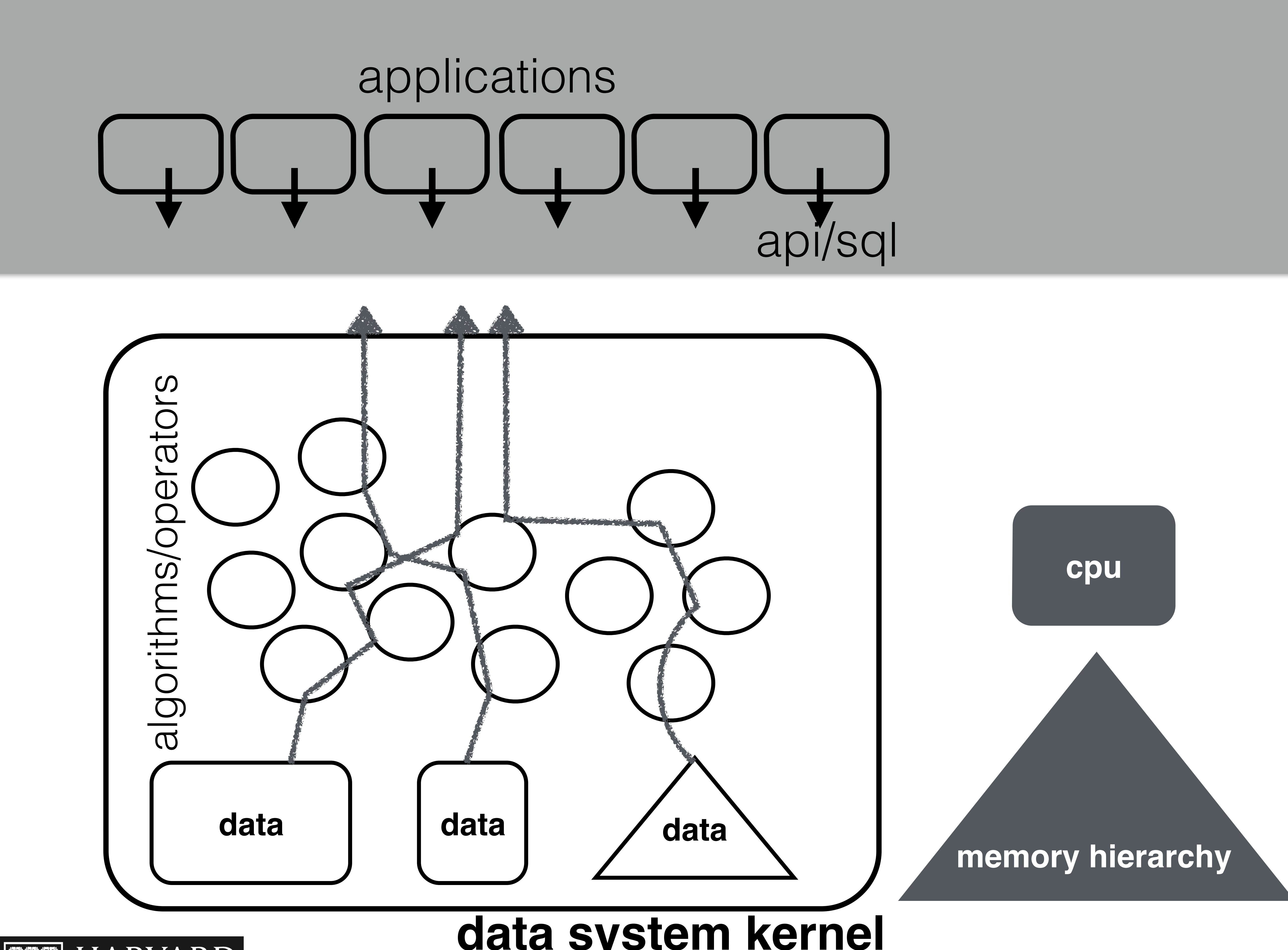
Data Systems that are Easy to Design, Tune and Use

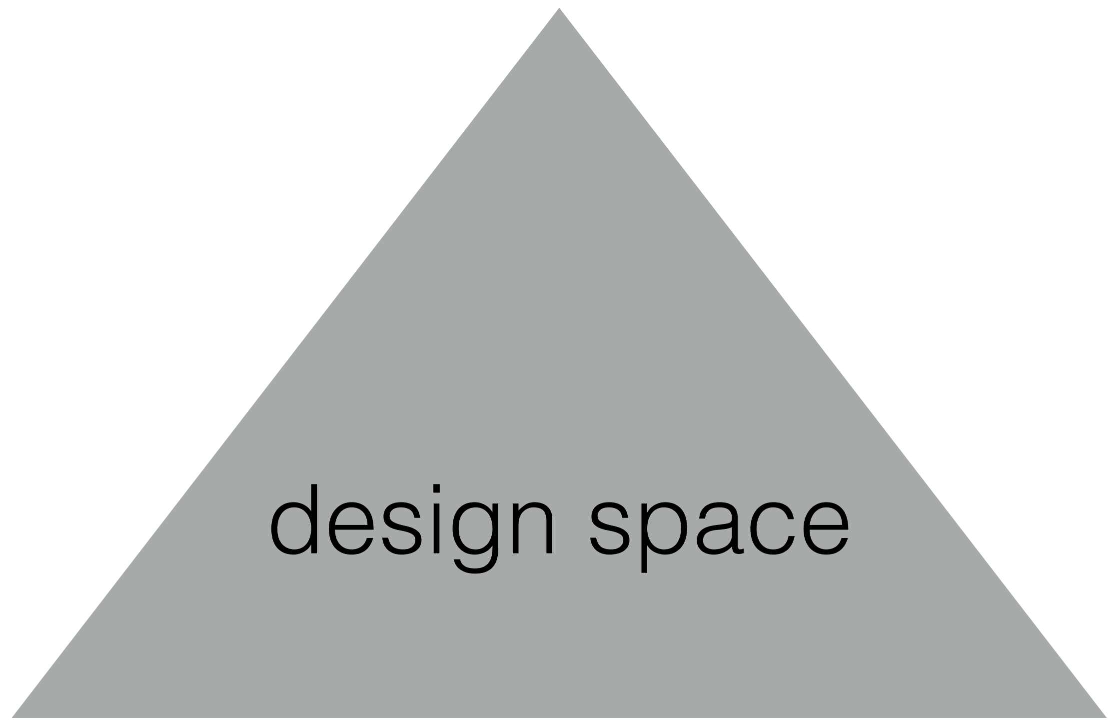
Stratos Idreos



DASlab @ Harvard SEAS

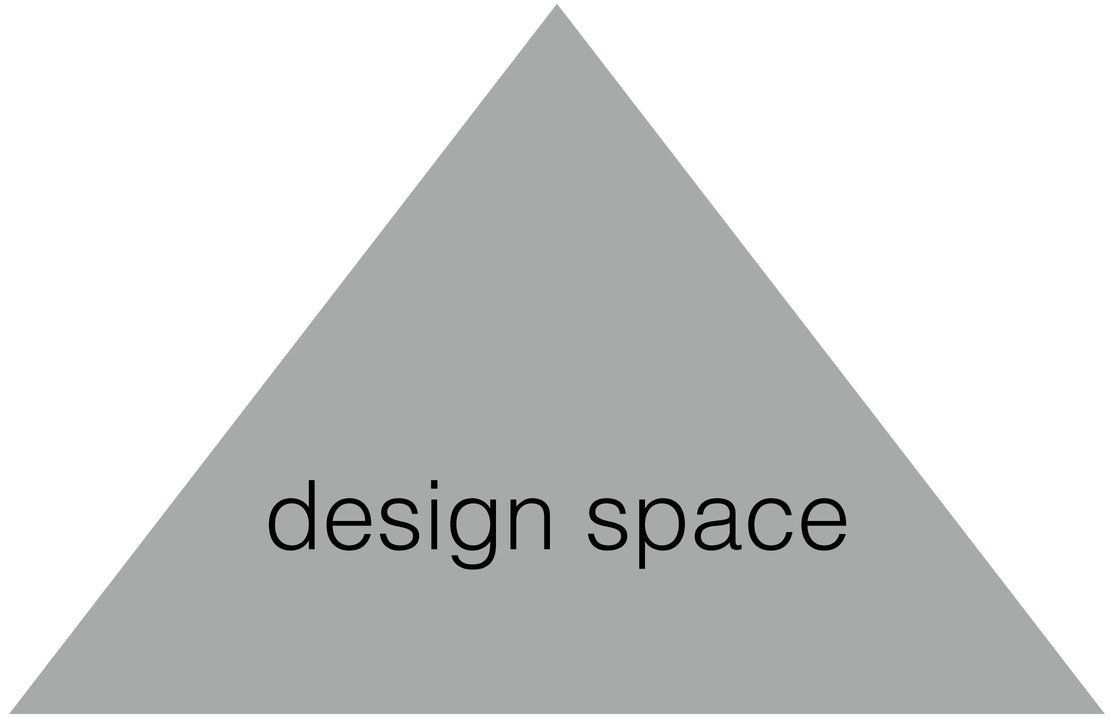
HARVARD School of Engineering and Applied Sciences


applications
api/sql
algorithms/operators
data
data
data
data system kernel
cpu
memory hierarchy



it all starts with how we store data

every bit matters



it all starts with how we store data

every bit matters

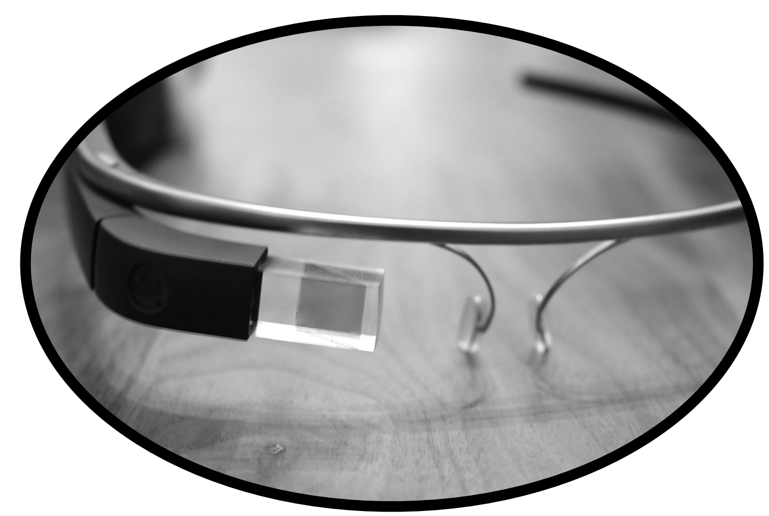
no fixed decisions

from static to dynamic designs

today

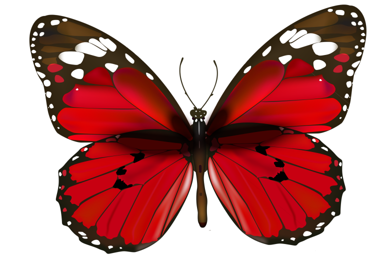
today

tomorrow



soon everyone will need to be a "data scientist"

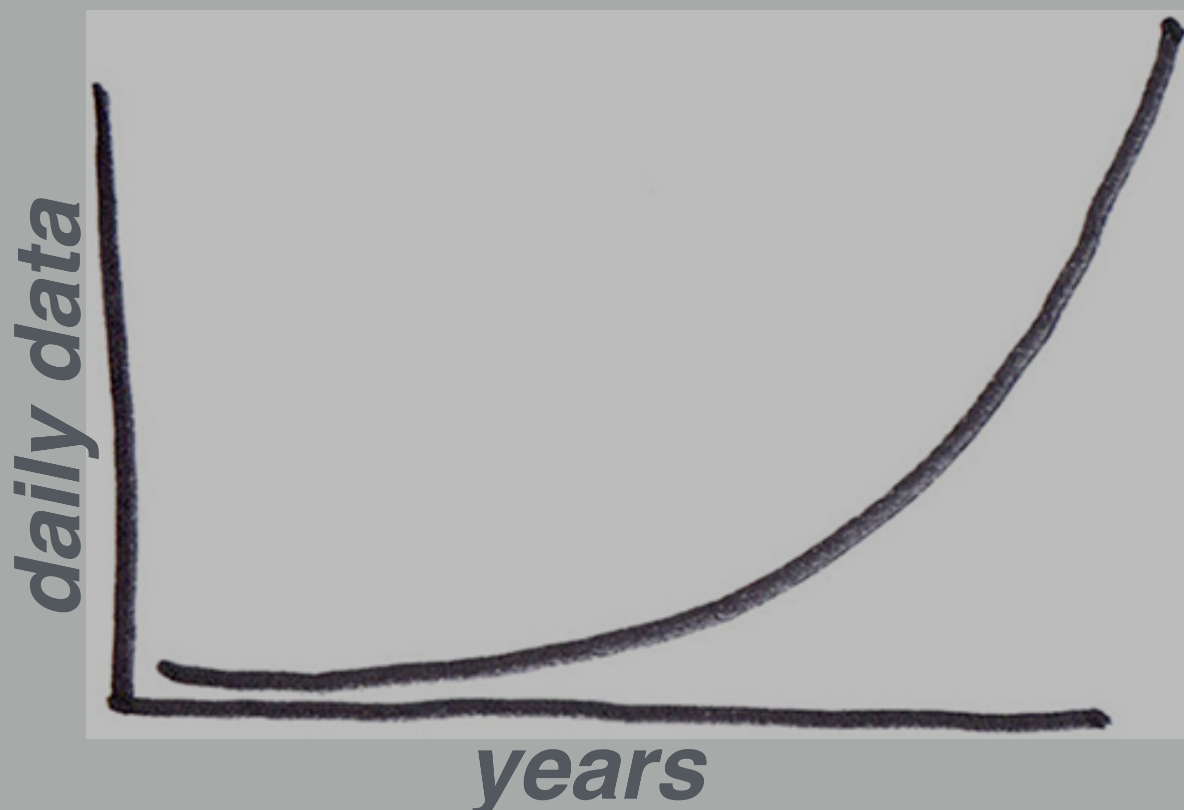
data exploration



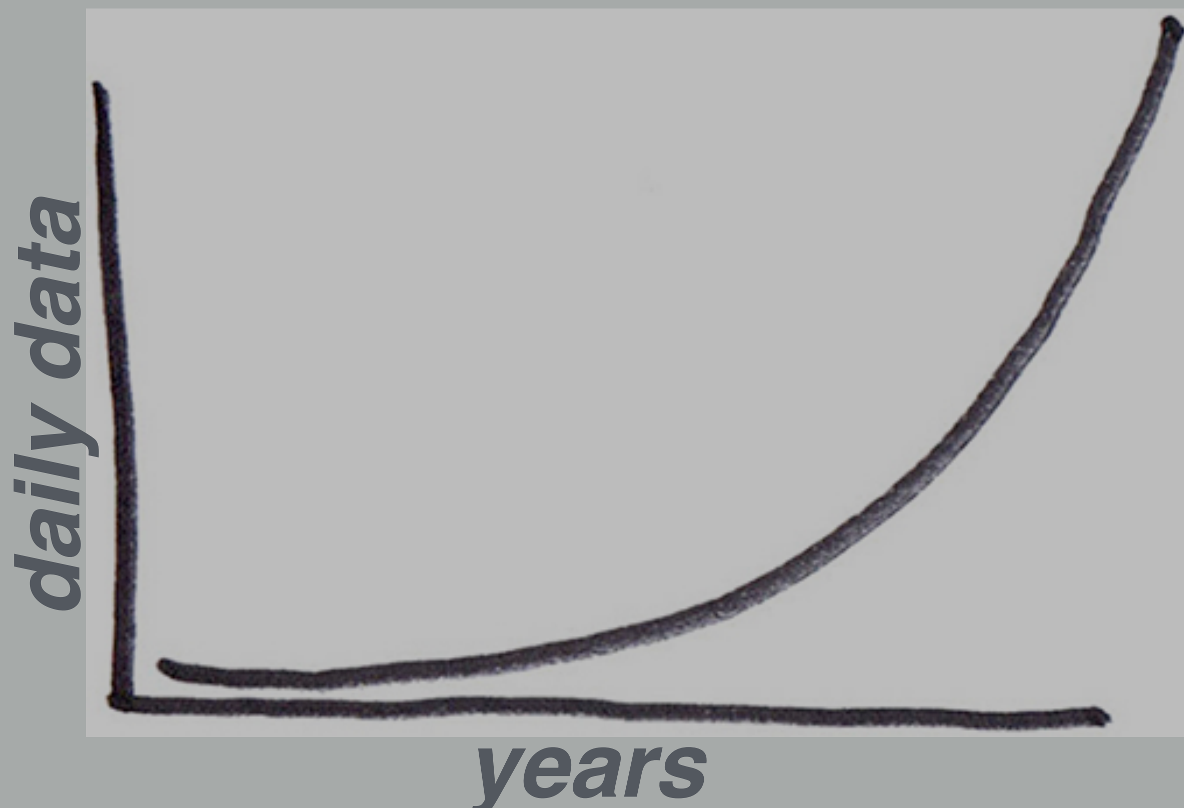
not always sure
what we are looking for
(until we find it)

data has always been **big**

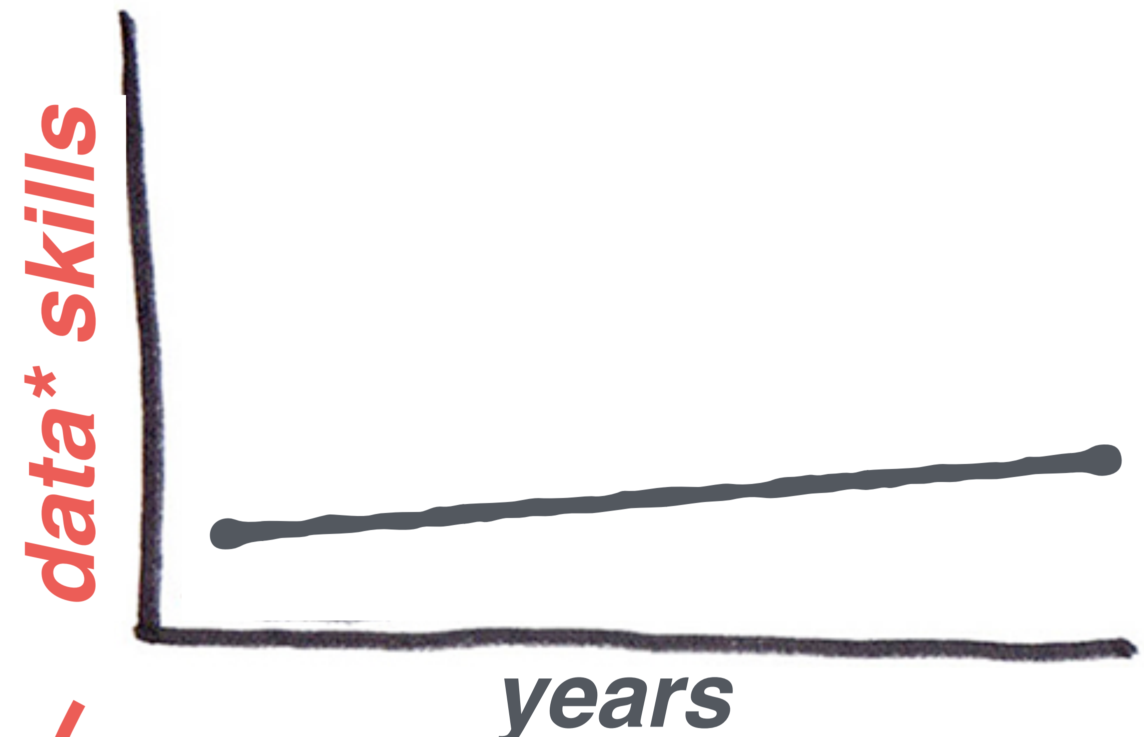
volume | velocity | variety | veracity



[IBMbigdata]



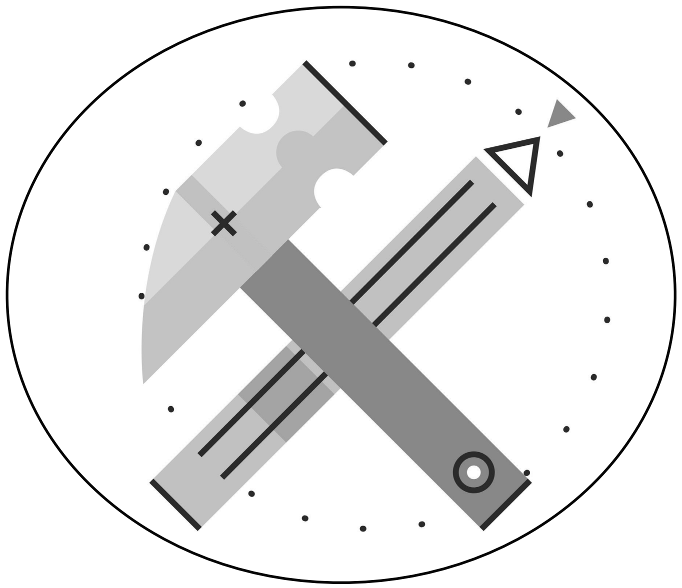
[IBMbigdata]



[StratosGuess]

data system design,
set-up, tune, use

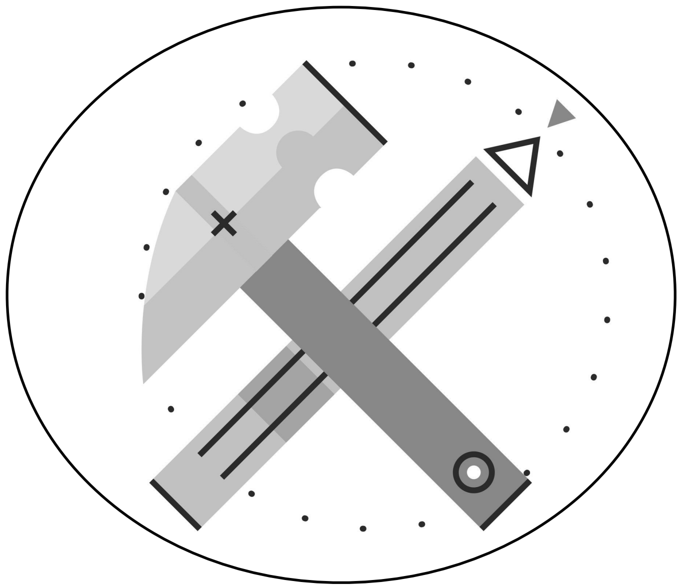
data systems that are **easy** to:



(years)

design & build

data systems that are **easy** to:



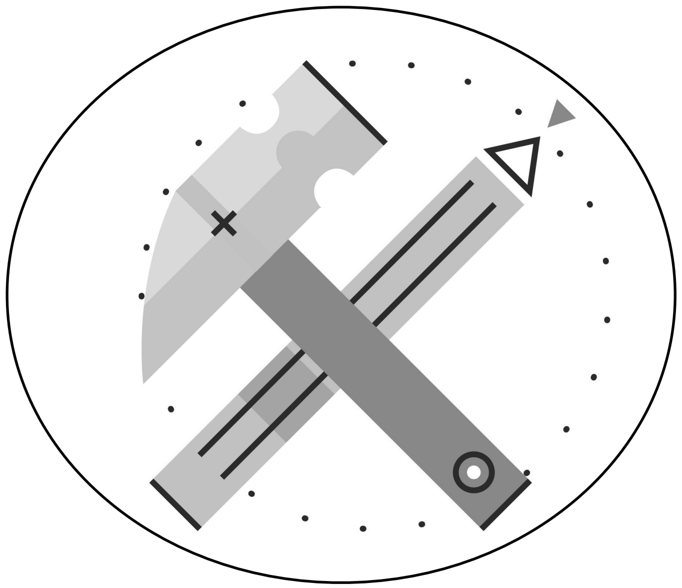
(years)

design & build

(months)

set-up & tune

data systems that are **easy** to:



(years)

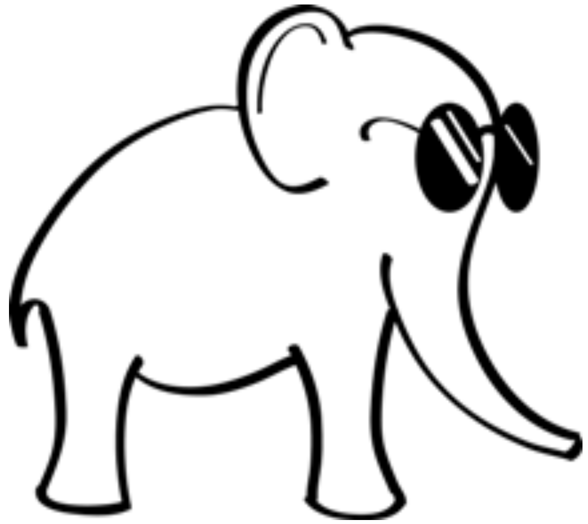
design & build

(months)

set-up & tune

(hours/days)

use



too many preparation options
lead to complex installation

schema storage load indexing | query

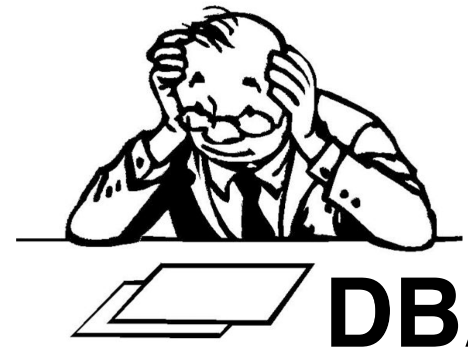
timeline →

expert users - idle time - workload knowledge

users/applications

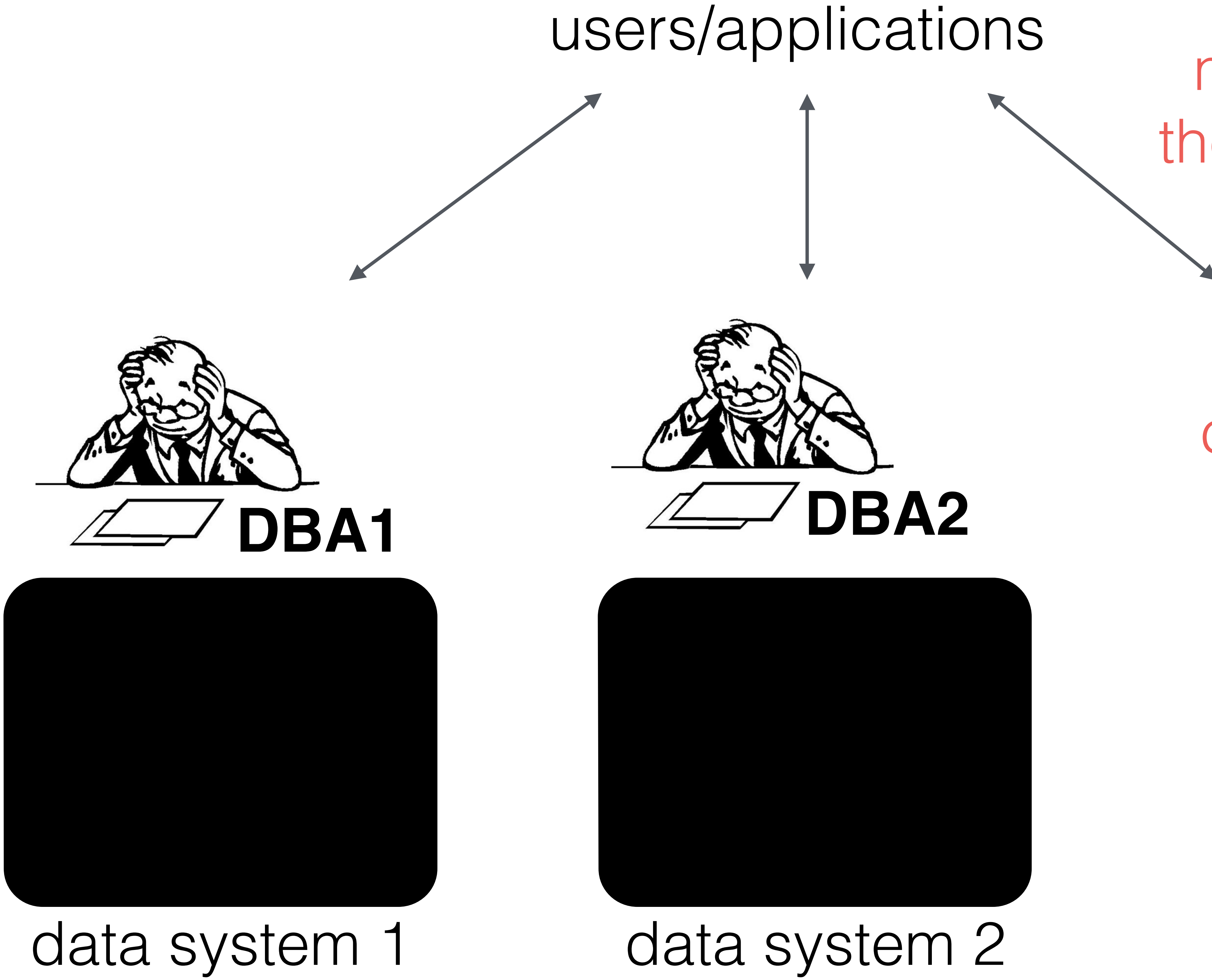
declarative interface

ask what you want



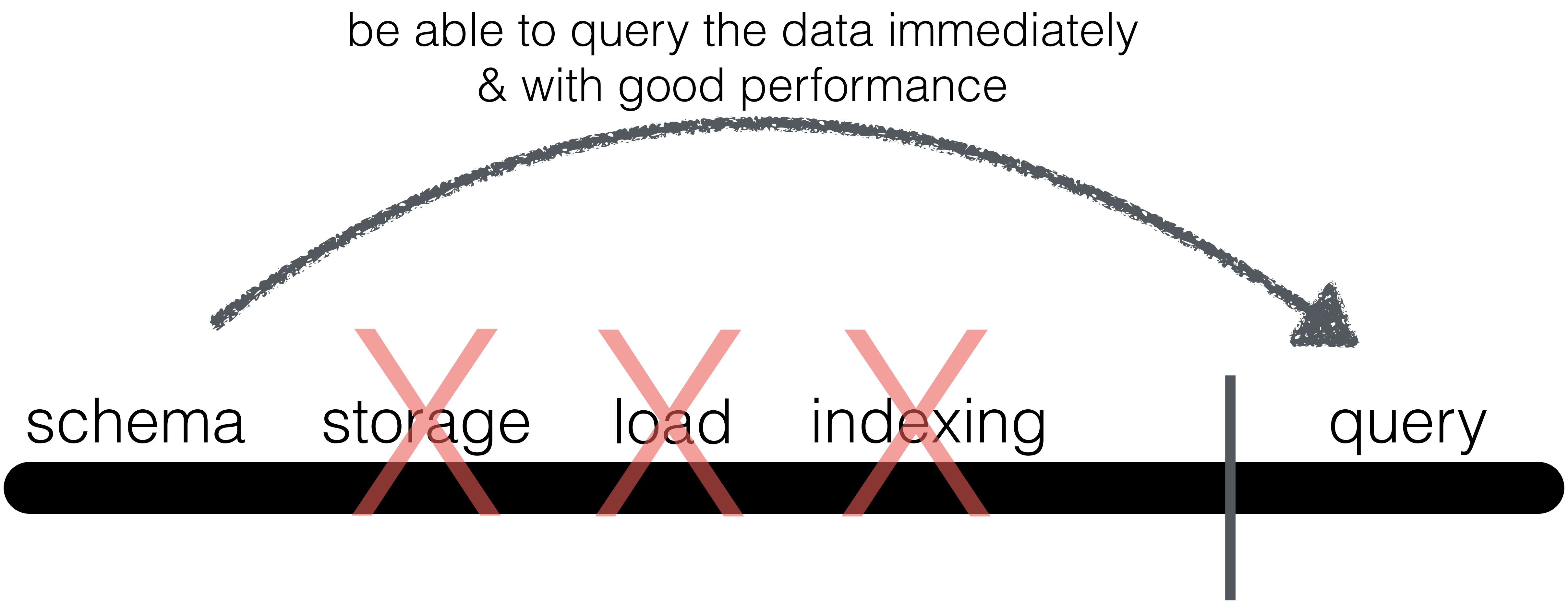
DBA

db system



need to choose the proper system & workloads/ applications change rapidly

...



HARVARD
School of Engineering
and Applied Sciences
Stratos Idreos

be able to query the data immediately
& with good performance

schema ~~storage~~ ~~load~~ ~~indexing~~ query

raw data → knowledge

indexing

storage load indexing query

tune= create proper indices offline

performance 10-100X

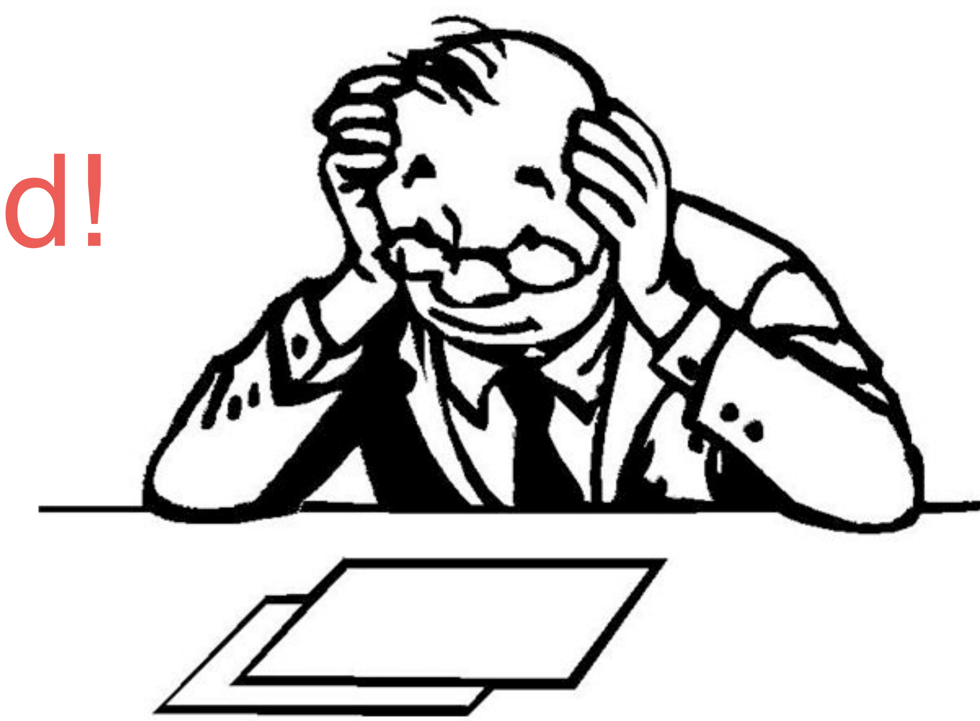
indexing

tune= create proper indices offline

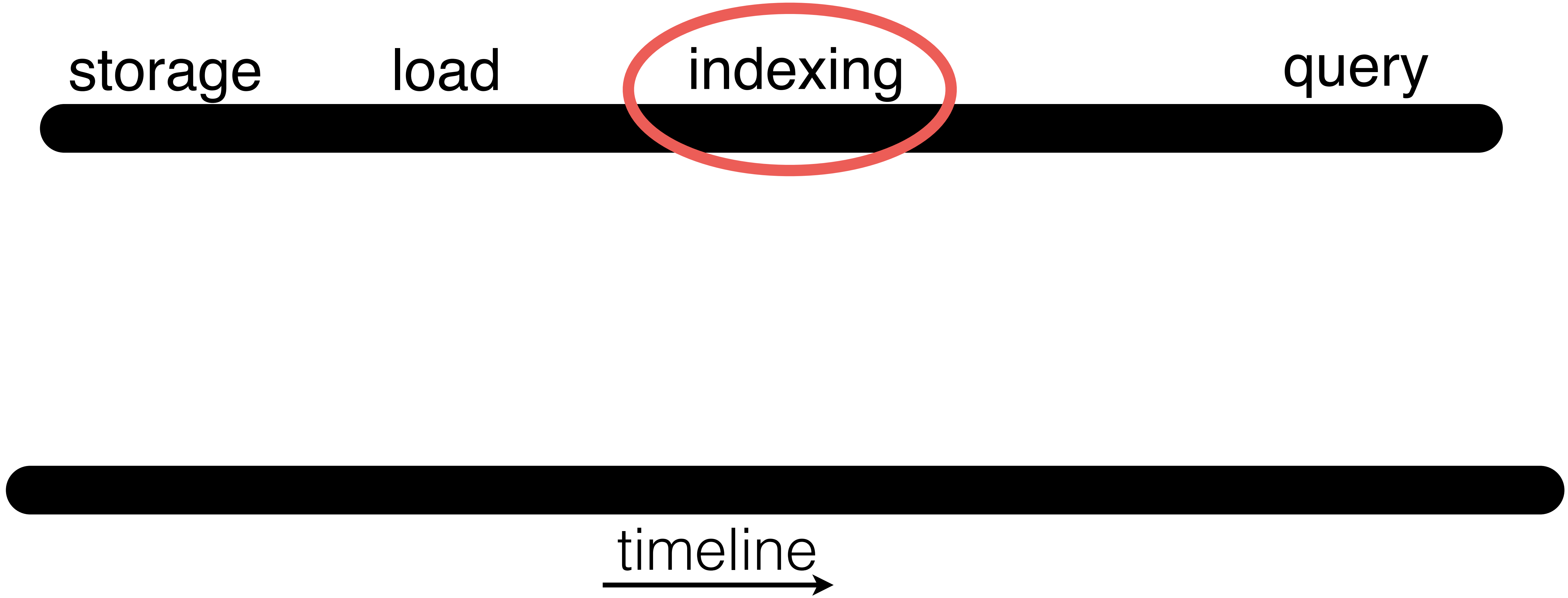
performance 10-100X

but it depends on workload!

which indices to build?
on which data parts?
and when to build them?



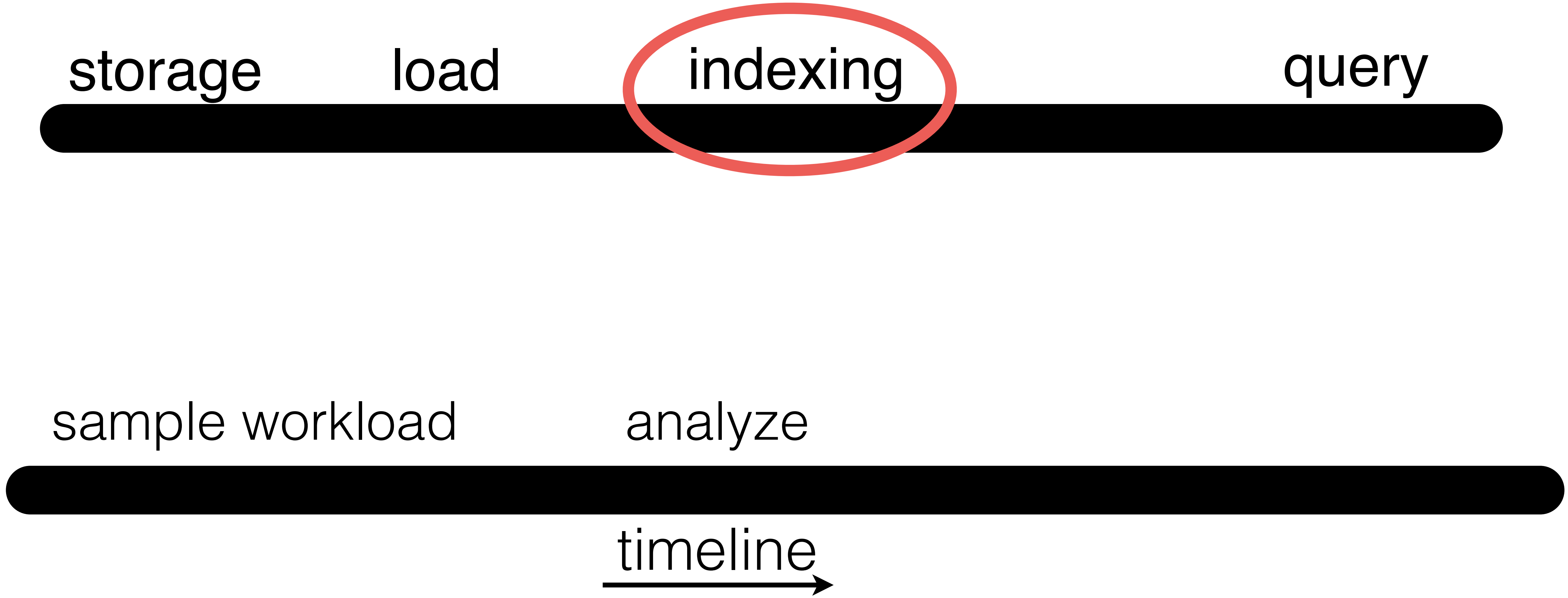
storage load indexing query

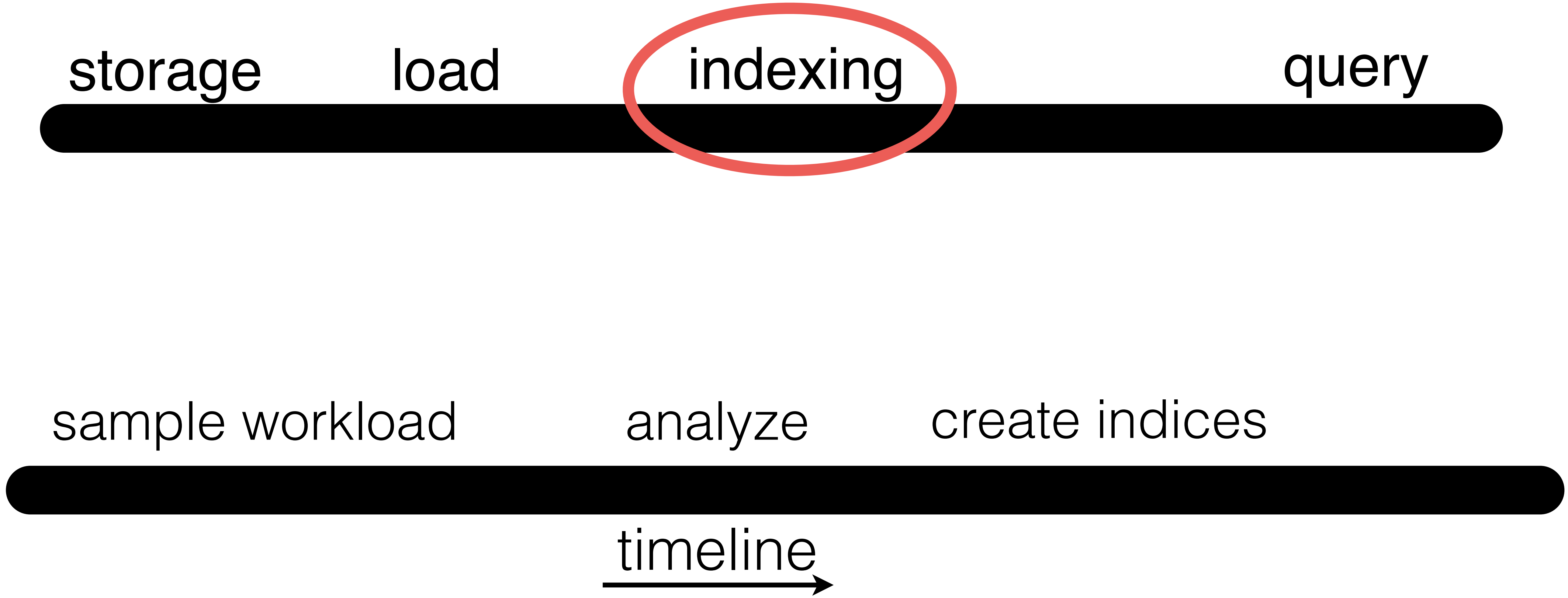

storage
load
indexing
query
timeline

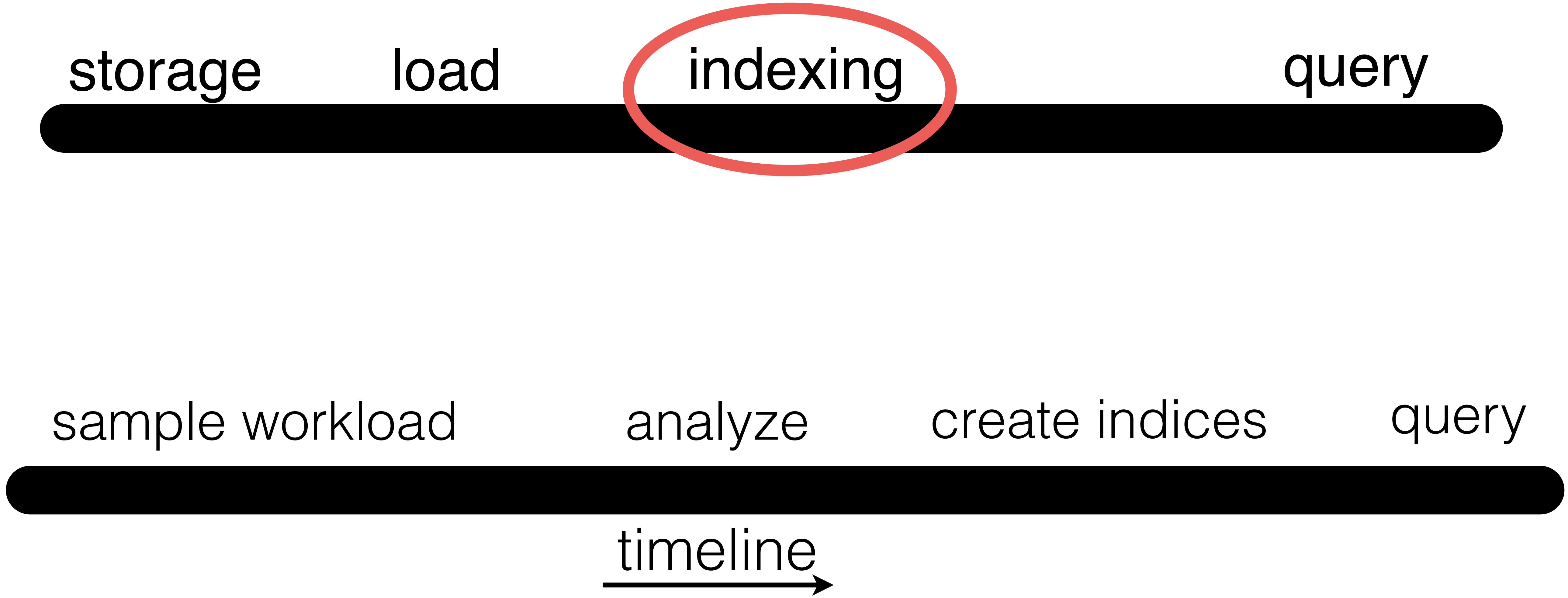
storage load indexing query

sample workload

timeline


storage
load
indexing
query
sample workload
analyze
timeline


storage
load
indexing
query
sample workload
analyze
create indices
timeline


storage
load
indexing
query
sample workload
analyze
create indices
query
timeline

storage load indexing query

sample workload analyze create indices query

timeline

complex and time consuming process

human administrators + auto-tuning tools

sample workload analyze create indices query

timeline →

complex and time consuming process

big data V's

volume

velocity

variety

veracity

what can go wrong?

not enough space to index all data

not enough idle **time** to finish proper tuning

by the time we finish tuning, the **workload changes**

not enough money - energy - resources big data V's

volume

velocity

variety

veracity

what can go wrong?

not enough space to index all data

not enough idle **time** to finish proper tuning

by the time we finish tuning, the **workload changes**

not enough money - energy - resources

database cracking

database cracking

~~idle time~~

~~workload knowledge~~

~~external tools~~

~~human control~~

database cracking

auto-tuning database kernels

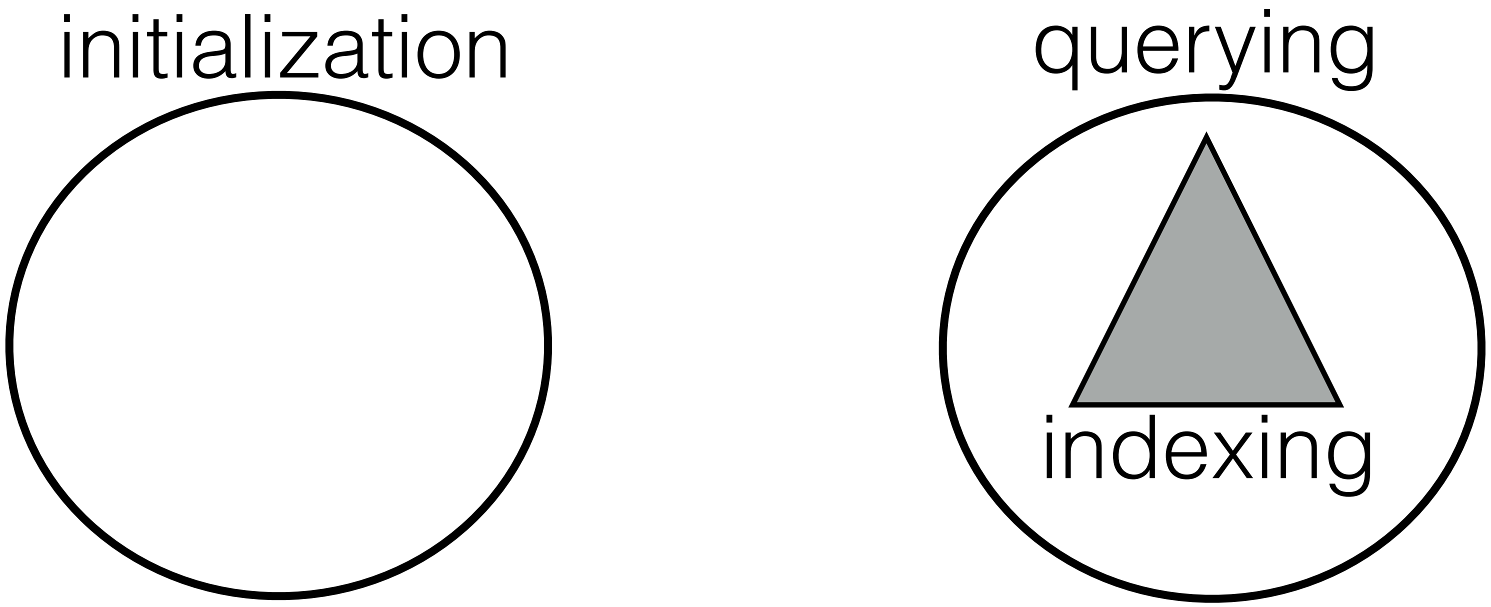
incremental, adaptive, partial indexing

~~idle time~~

~~workload knowledge~~

~~external tools~~

~~human control~~



database cracking

auto-tuning database kernels

incremental, adaptive, partial indexing

~~idle time~~

~~workload knowledge~~

~~external tools~~

~~human control~~

initialization

querying

indexing

database cracking

auto-tuning database kernels

incremental, adaptive, partial indexing

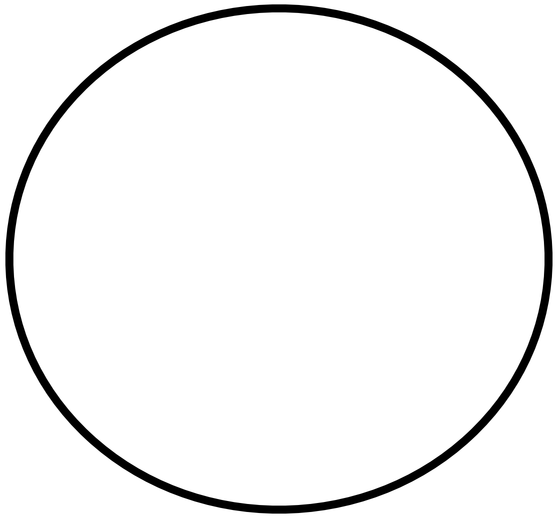
~~idle time~~

~~workload knowledge~~

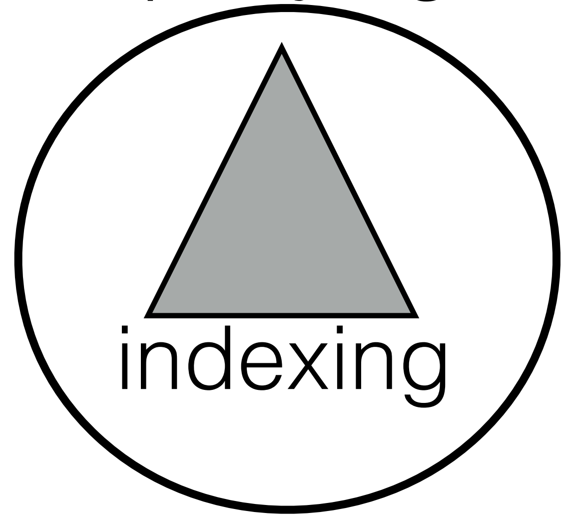
~~external tools~~

~~human control~~

initialization



querying



database cracking

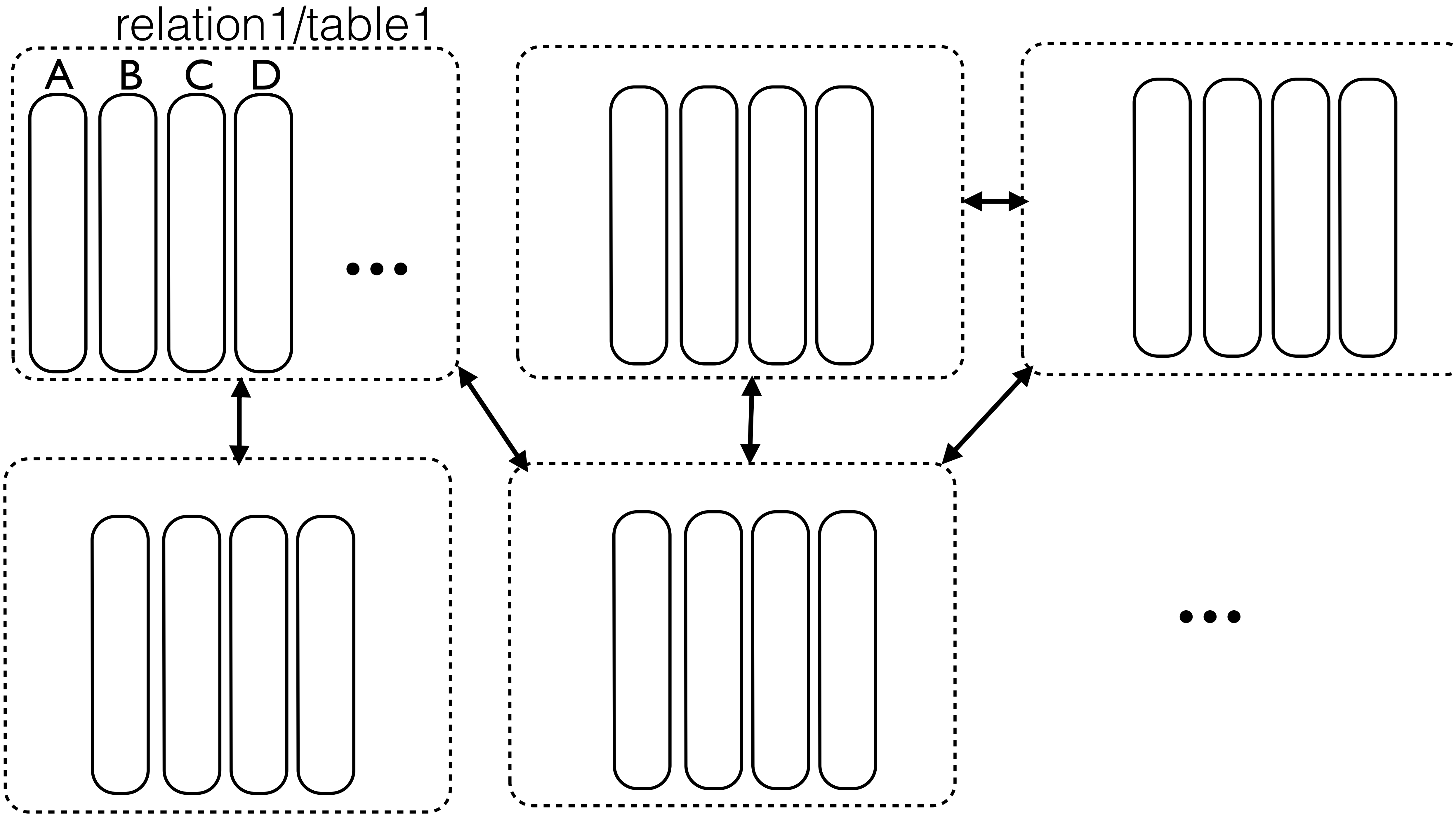
auto-tuning database kernels

incremental, adaptive, partial indexing

every query is treated as an advice on how data should be stored

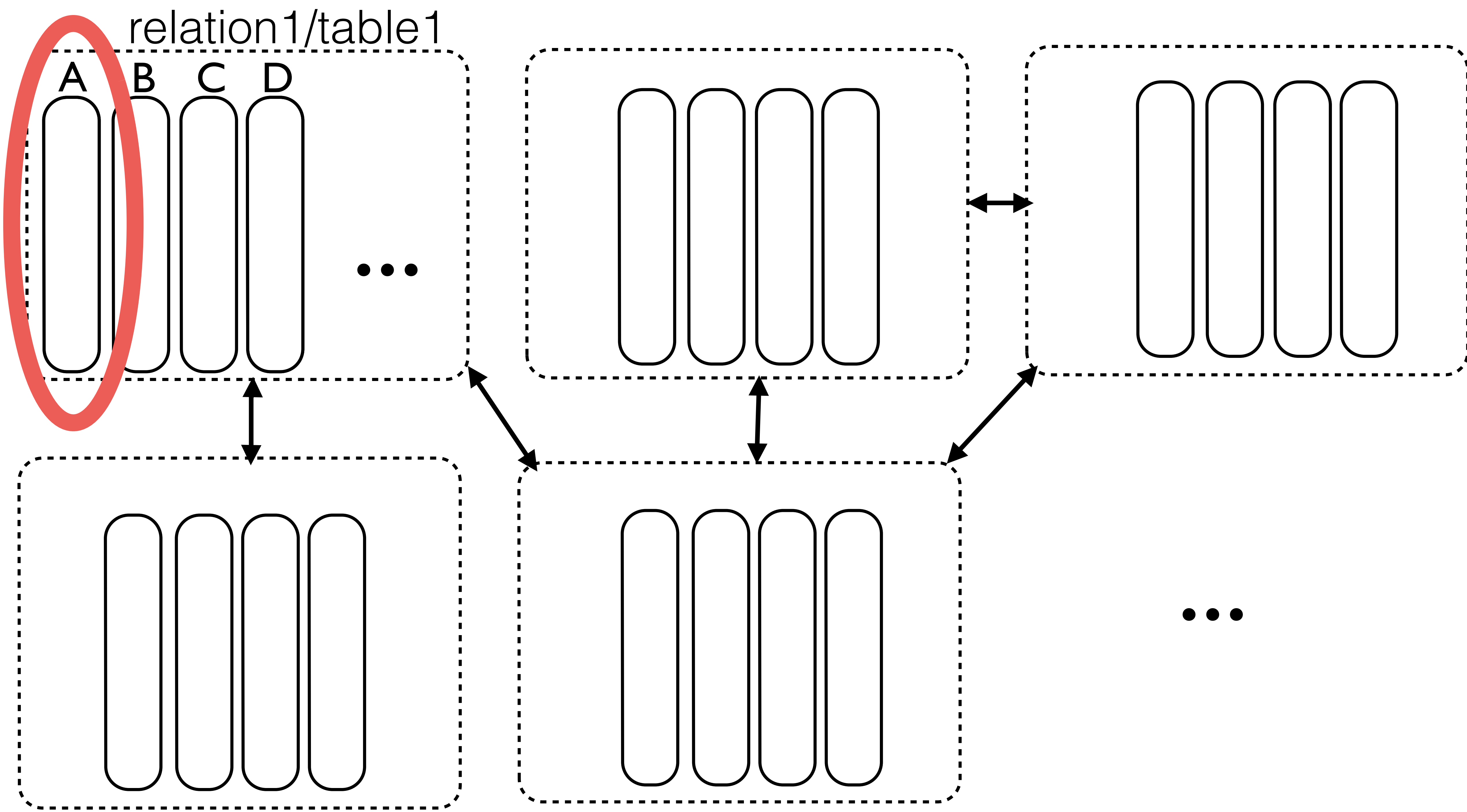
column-store database

a fixed-width and dense array per attribute



column-store database

a fixed-width and dense array per attribute



column A

Q1:
select R.A
from R
where R.A>10
and R.A<14

column A
13
16
4
9
2
12
7
1
19
3
14
11
8
6

column A

Q1:
select R.A
from R
where R.A>10
and R.A<14

13
16
4
9
2
12
7
1
19
3
14
11
8
6

Q1:
select R.A
from R
where R.A>10
and R.A<14

column A

column A		sorted
13		1
16		2
4		3
9		4
2	sort →	6
12		7
7		8
1		9
19		11
3		12
14		13
11		14
8		16
6		19

Q1:
select R.A
from R
where R.A>10
and R.A<14

column A

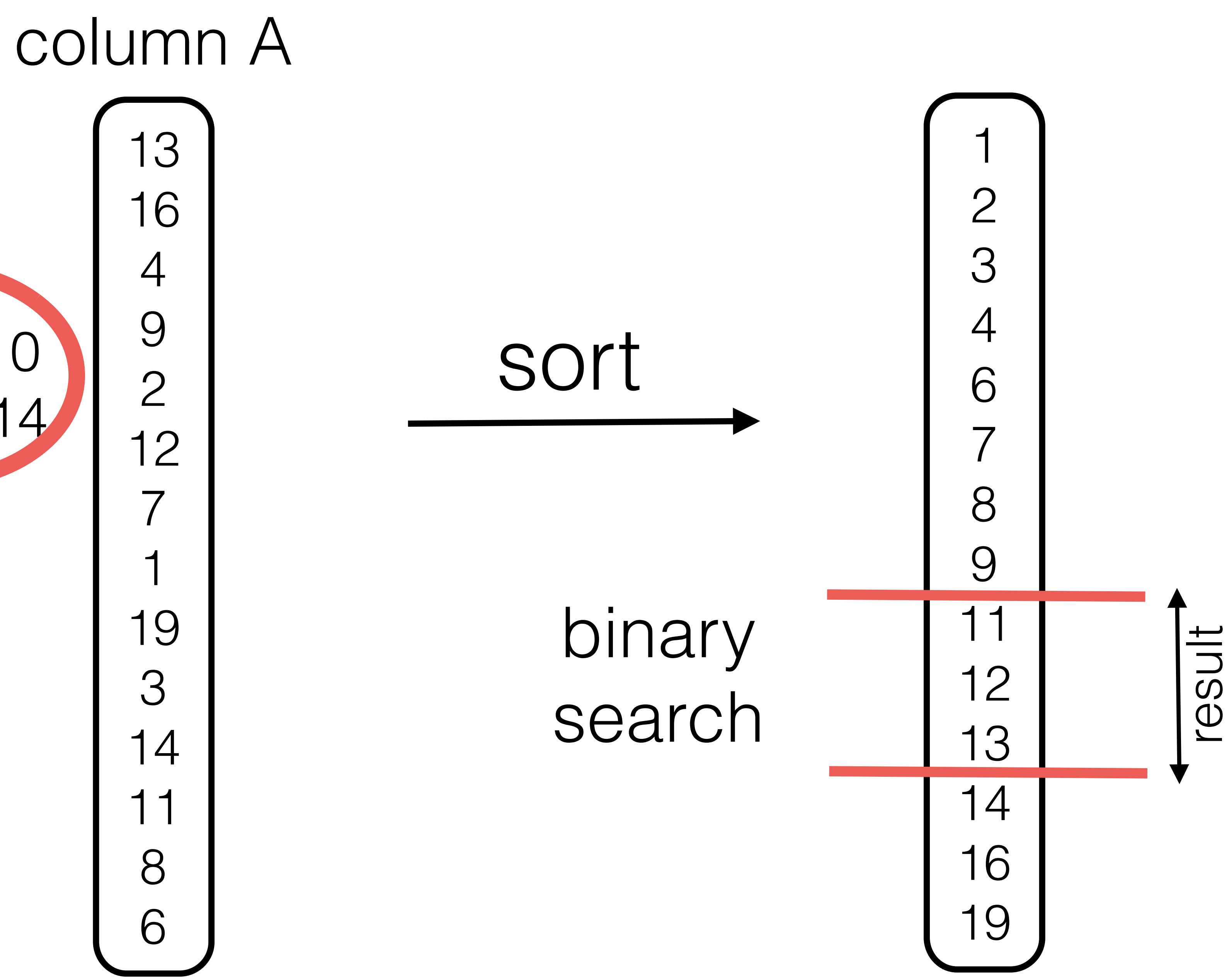
13
16
4
9
2
12
7
1
19
3
14
11
8
6

sort

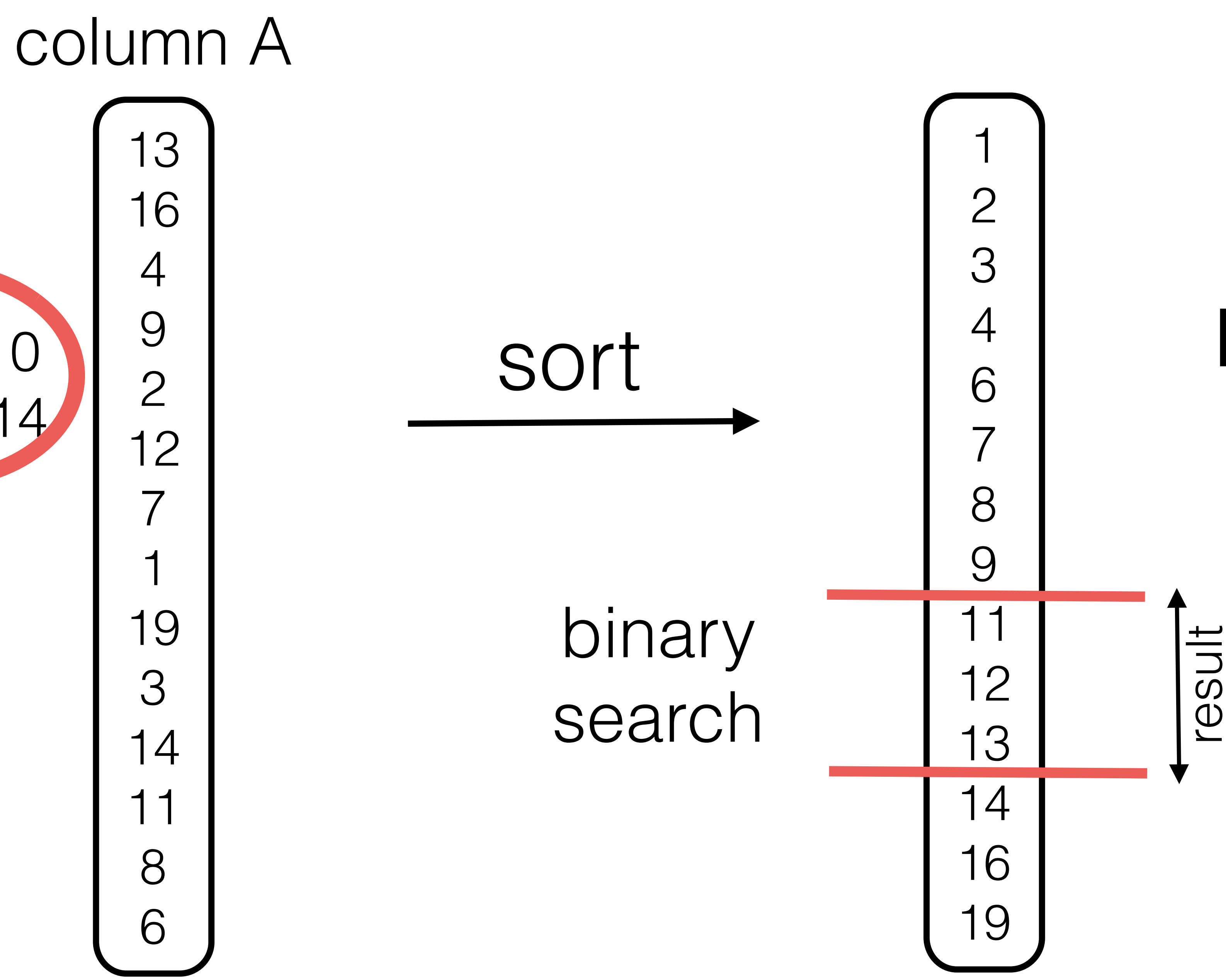
binary search

1
2
3
4
6
7
8
9
11
12
13
14
16
19

Q1:
select R.A
from R
where R.A>10
and R.A<14



Q1:
select R.A
from R
where R.A>10
and R.A<14



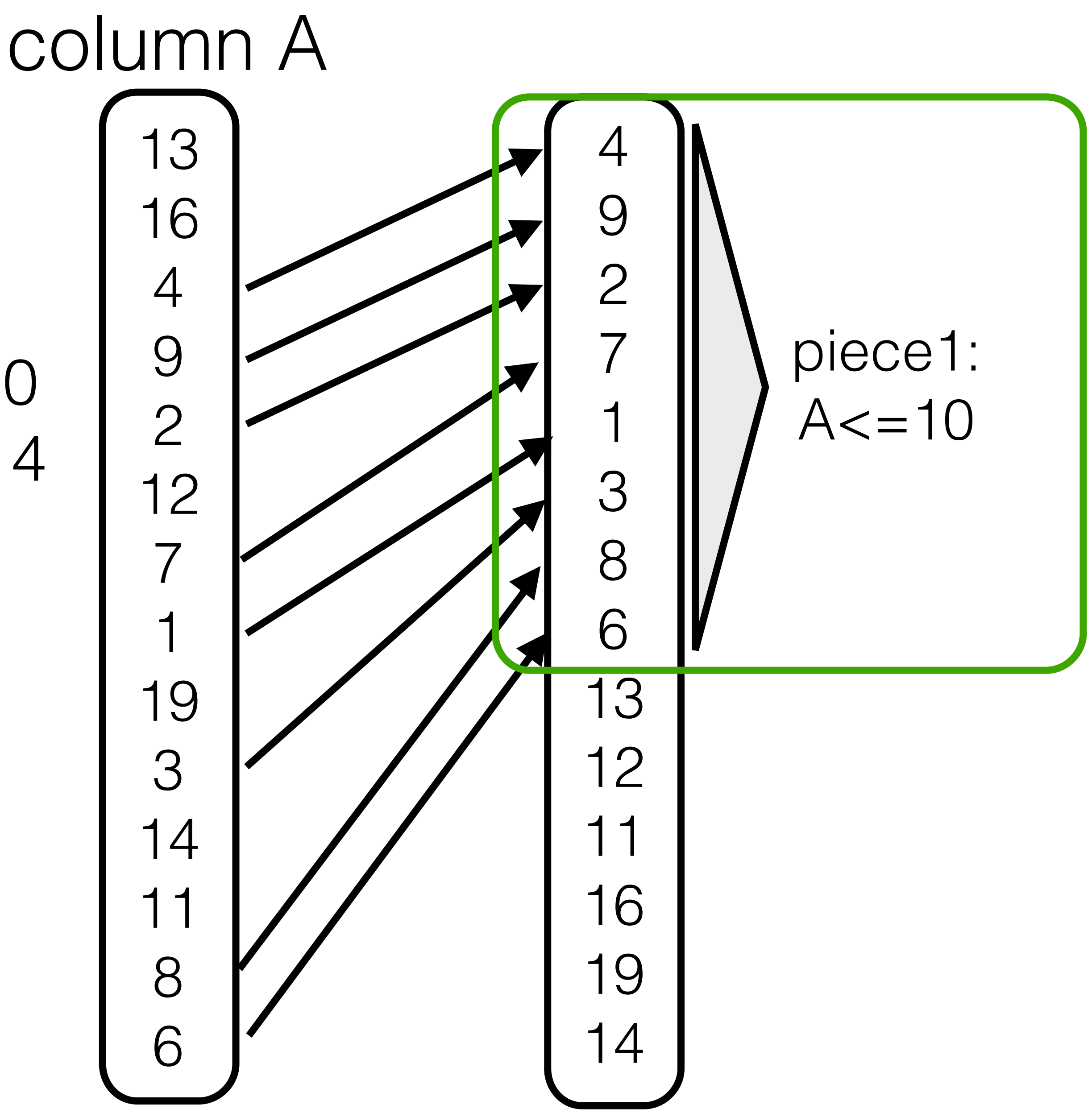
time
+
knowledge

column A

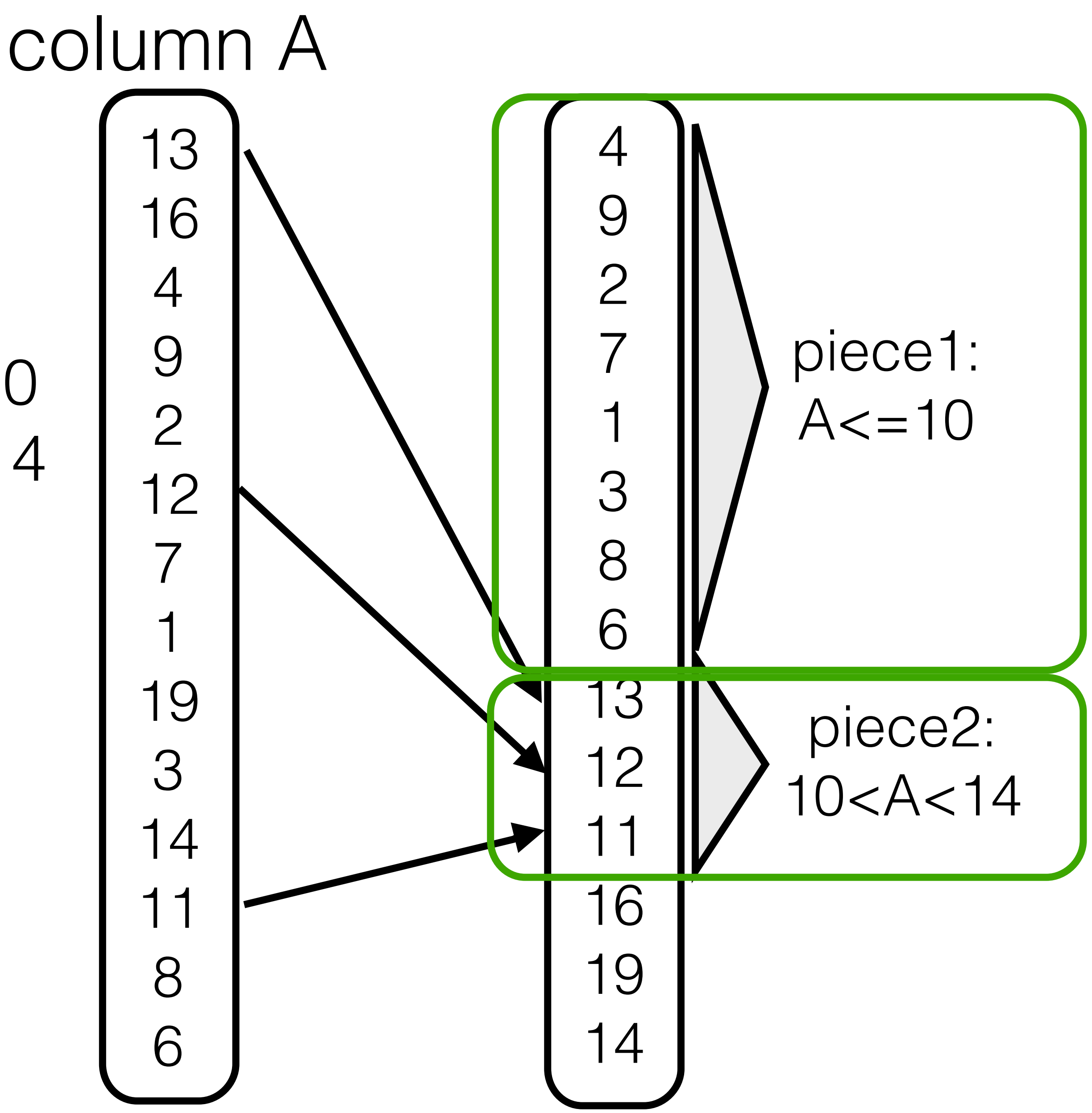
Q1:
select R.A
from R
where R.A>10
and R.A<14

13
16
4
9
2
12
7
1
19
3
14
11
8
6

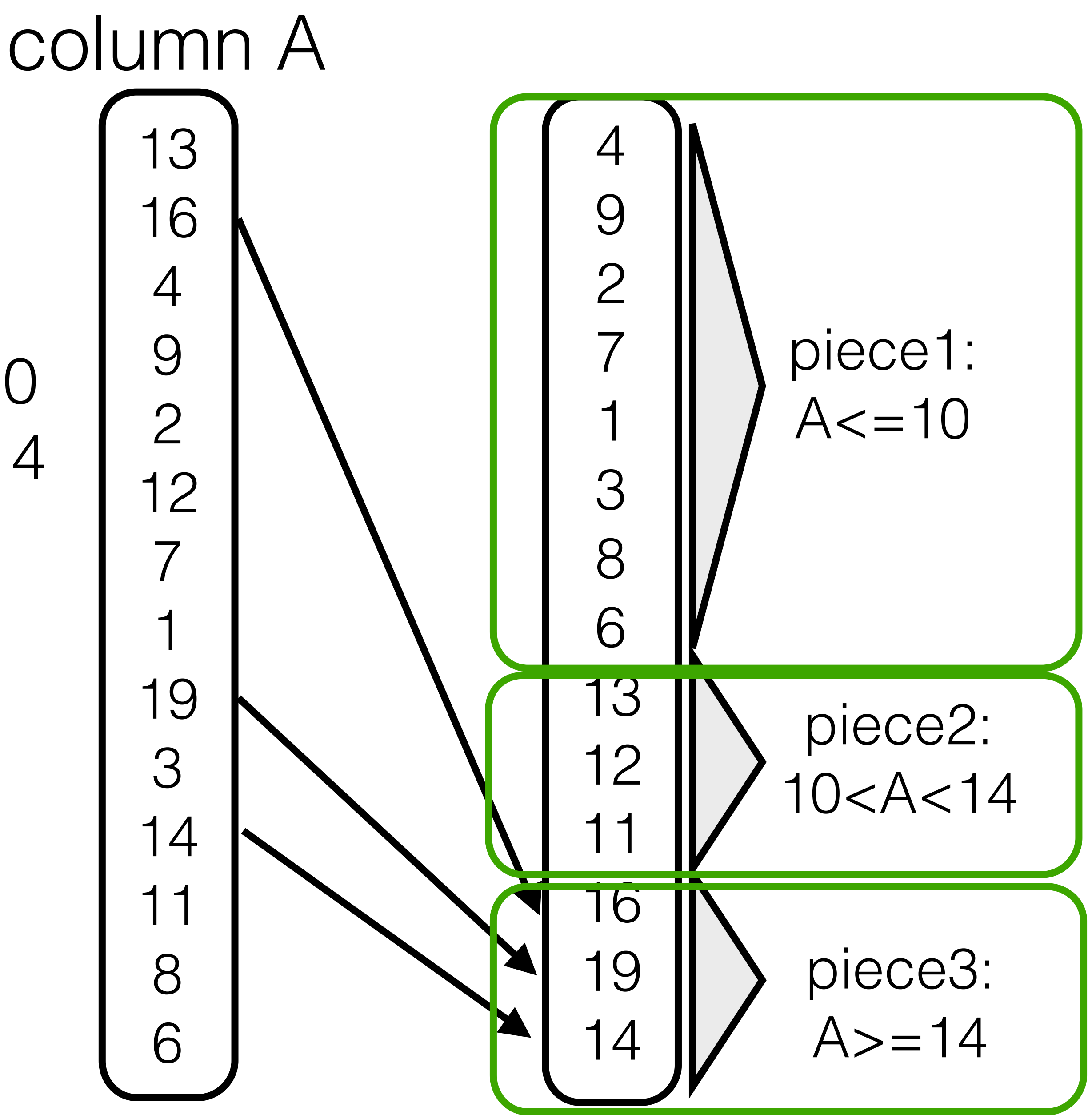
Q1:
select R.A
from R
where R.A>10
and R.A<14



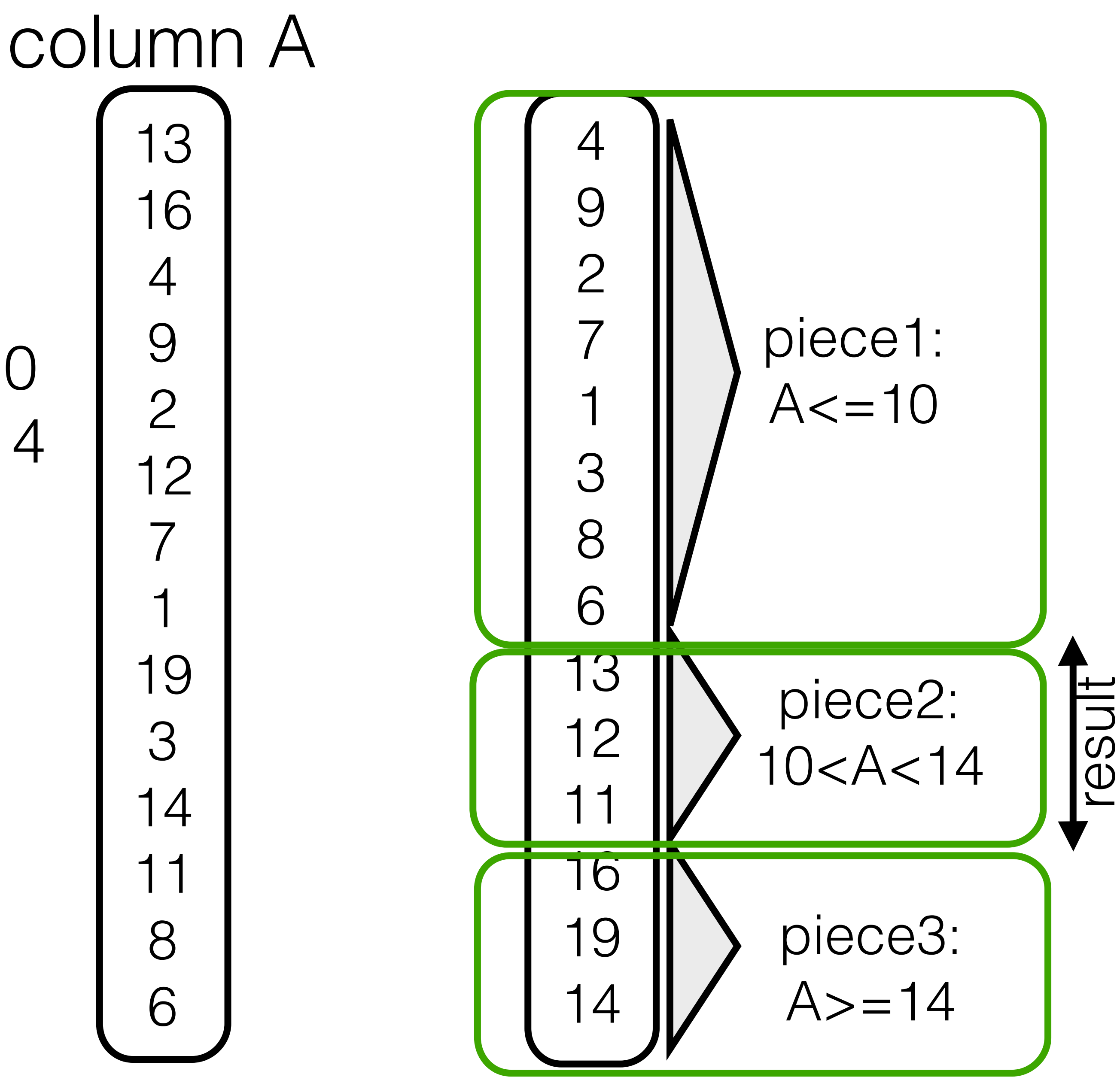
Q1:
select R.A
from R
where R.A>10
and R.A<14



Q1:
select R.A
from R
where R.A>10
and R.A<14

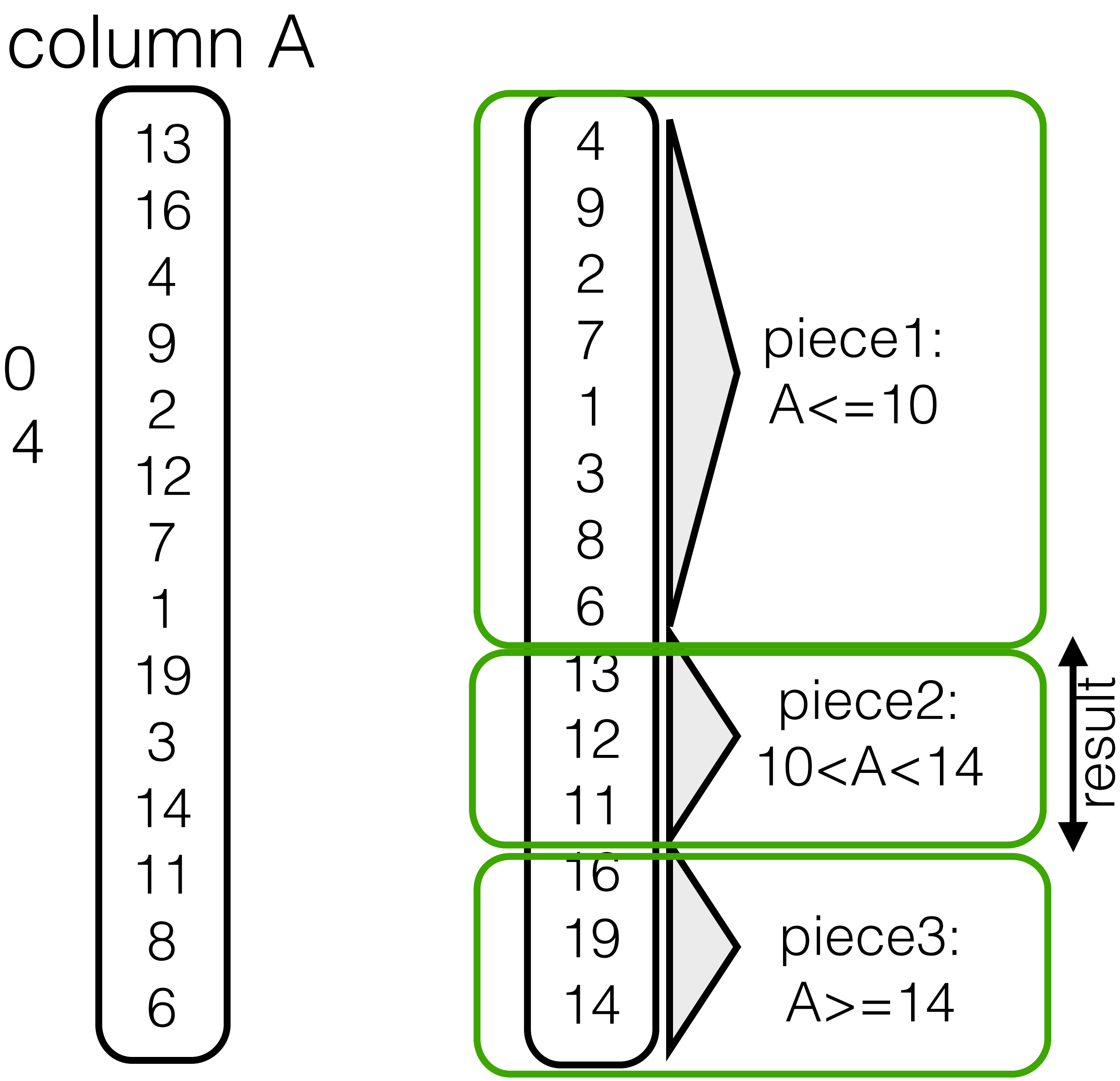


Q1:
select R.A
from R
where R.A>10
    and R.A<14



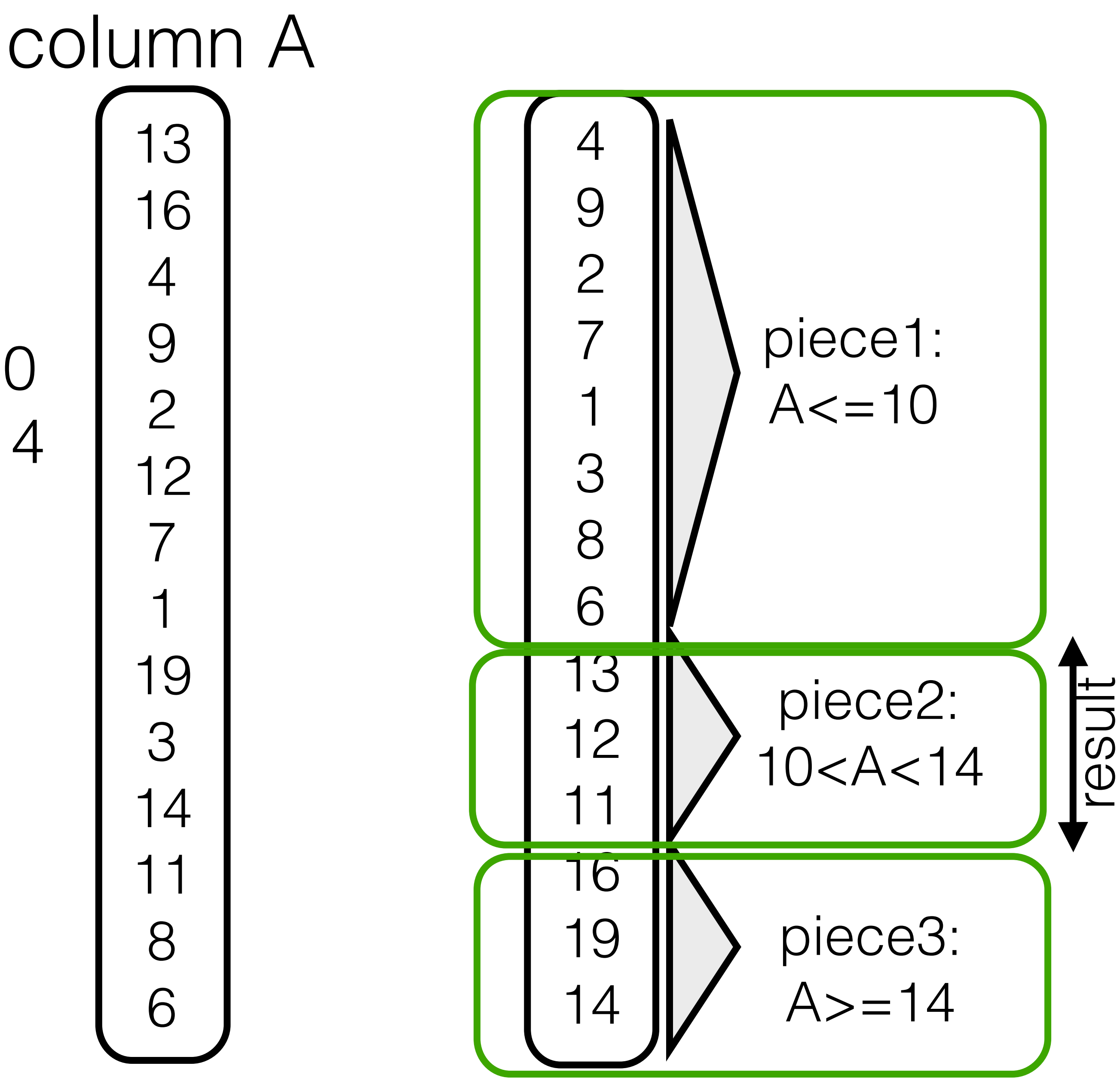
gain knowledge on how data is organized

Q1:
select R.A
from R
where R.A>10
and R.A<14



gain knowledge on how data is organized

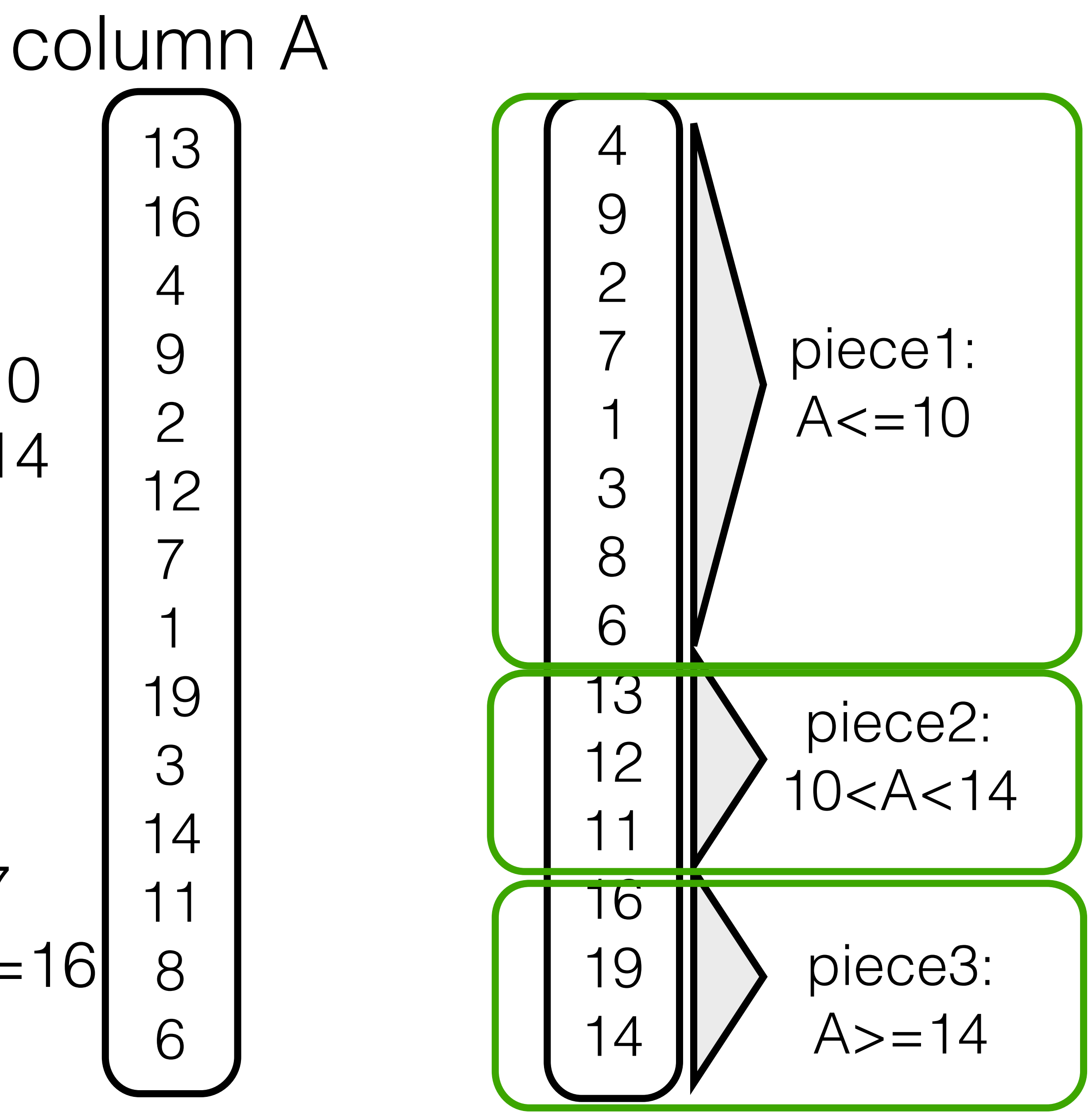
Q1:
select R.A
from R
where R.A>10
and R.A<14



dynamically/on-the-fly within the select-operator

Q1:
select R.A
from R
where R.A>10
and R.A<14

Q2:
select R.A
from R
where R.A>7
and R.A<=16



dynamically/on-the-fly within the select-operator

Q1:
select R.A
from R
where R.A>10
and R.A<14

Q2:
select R.A
from R
where R.A>7
and R.A<=16

column A

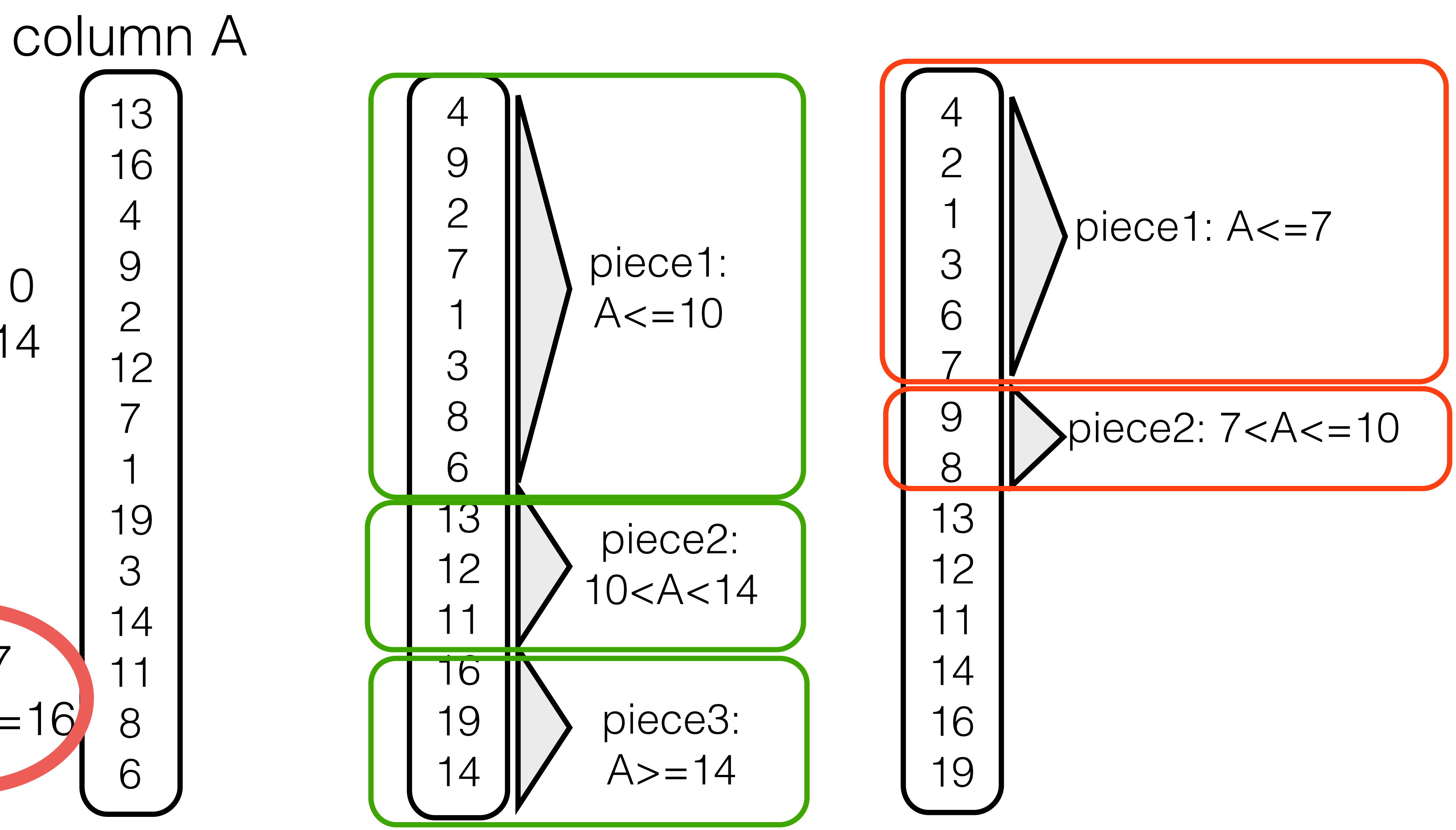
column A
13
16
4
9
2
12
7
1
19
3
14
11
8
6

4	piece1: A<=10
9	
2	
7	
1	
3	
8	
6	
13	piece2: 10<A<14
12	
11	
16	piece3: A>=14
19	
14	

dynamically/on-the-fly within the select-operator

Q1:
select R.A
from R
where R.A>10
and R.A<14

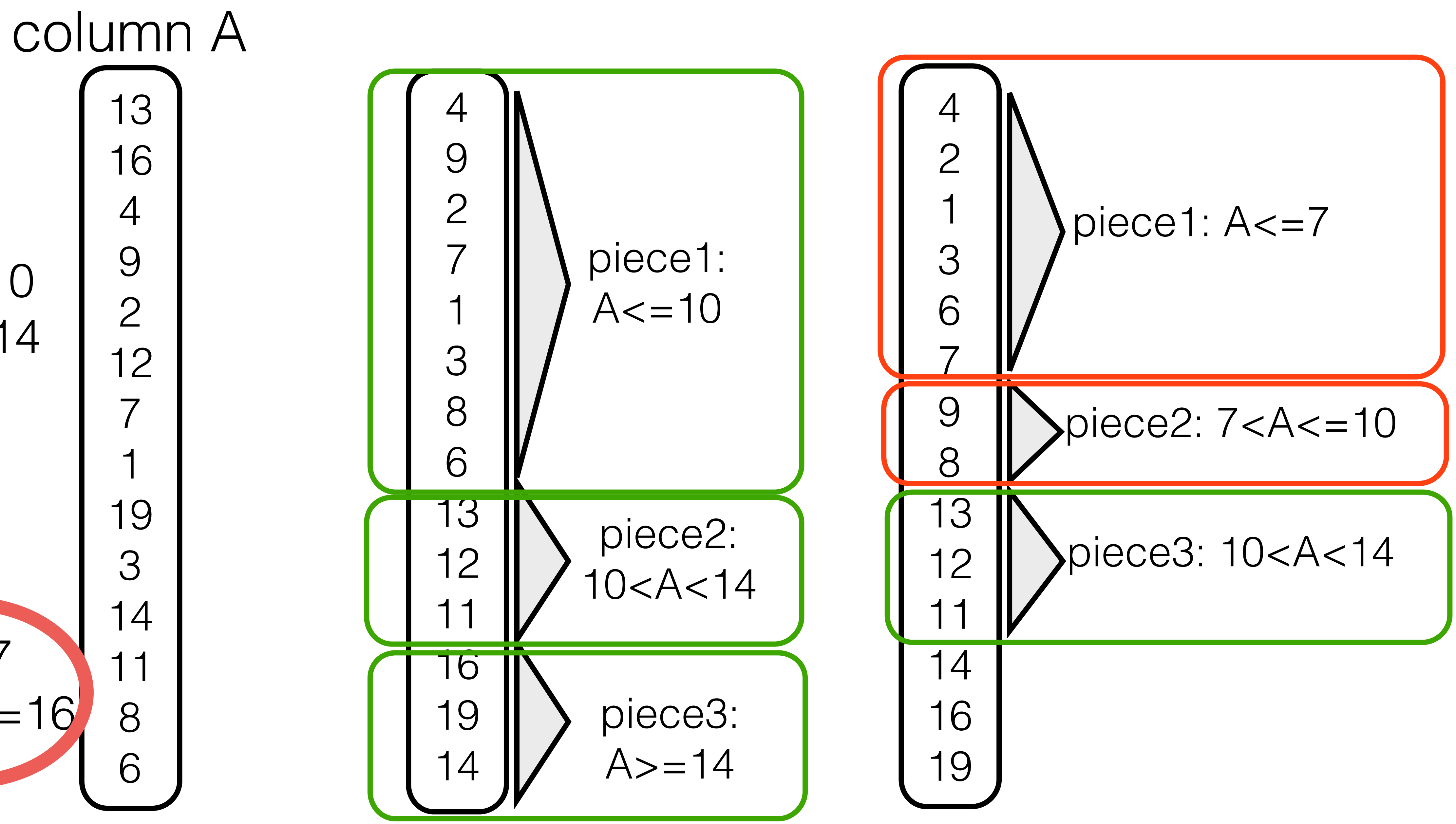
Q2:
select R.A
from R
where R.A>7
and R.A<=16



dynamically/on-the-fly within the select-operator

Q1:
select R.A
from R
where R.A>10
and R.A<14

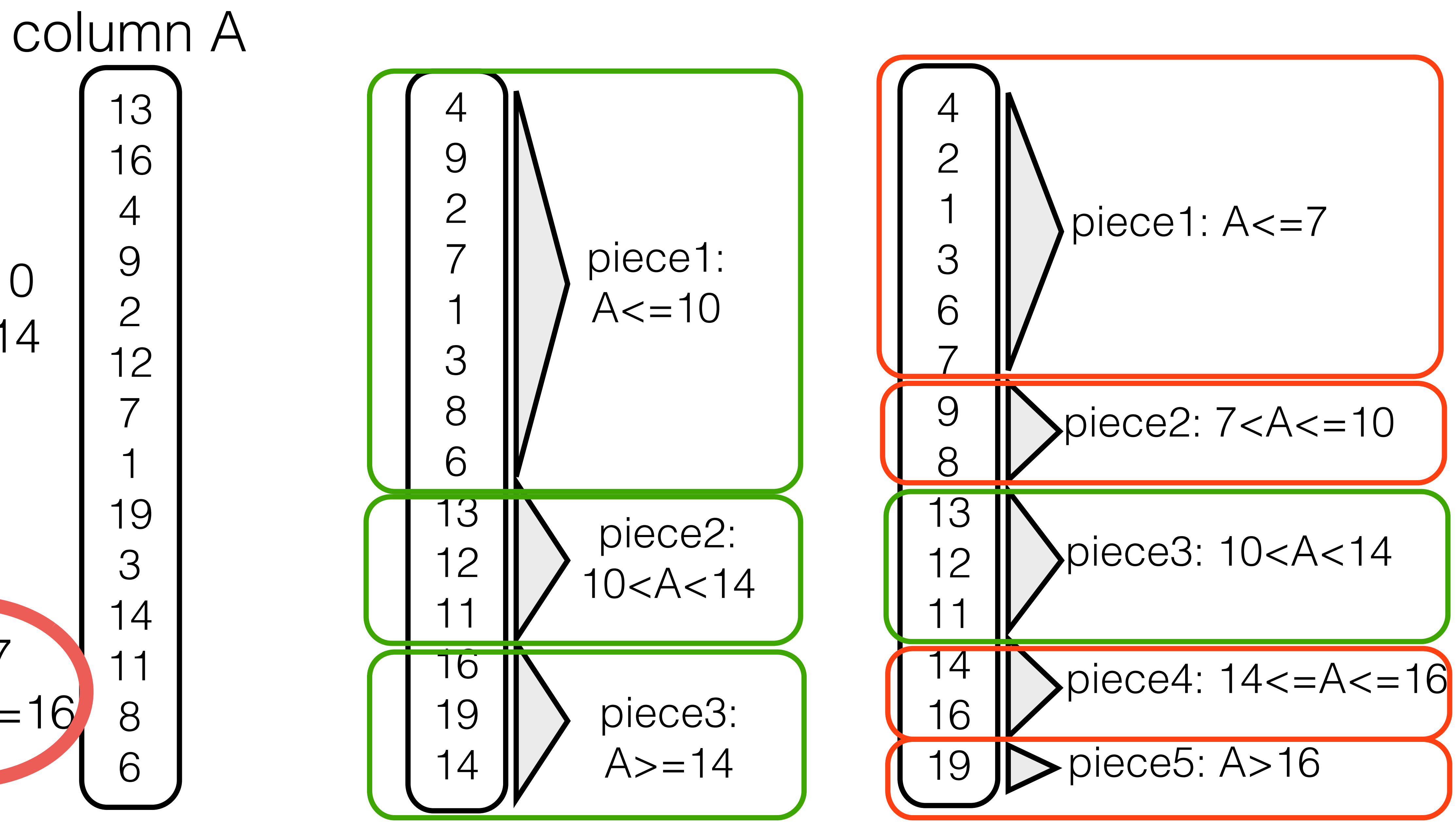
Q2:
select R.A
from R
where R.A>7
and R.A<=16



dynamically/on-the-fly within the select-operator

Q1:
select R.A
from R
where R.A>10
and R.A<14

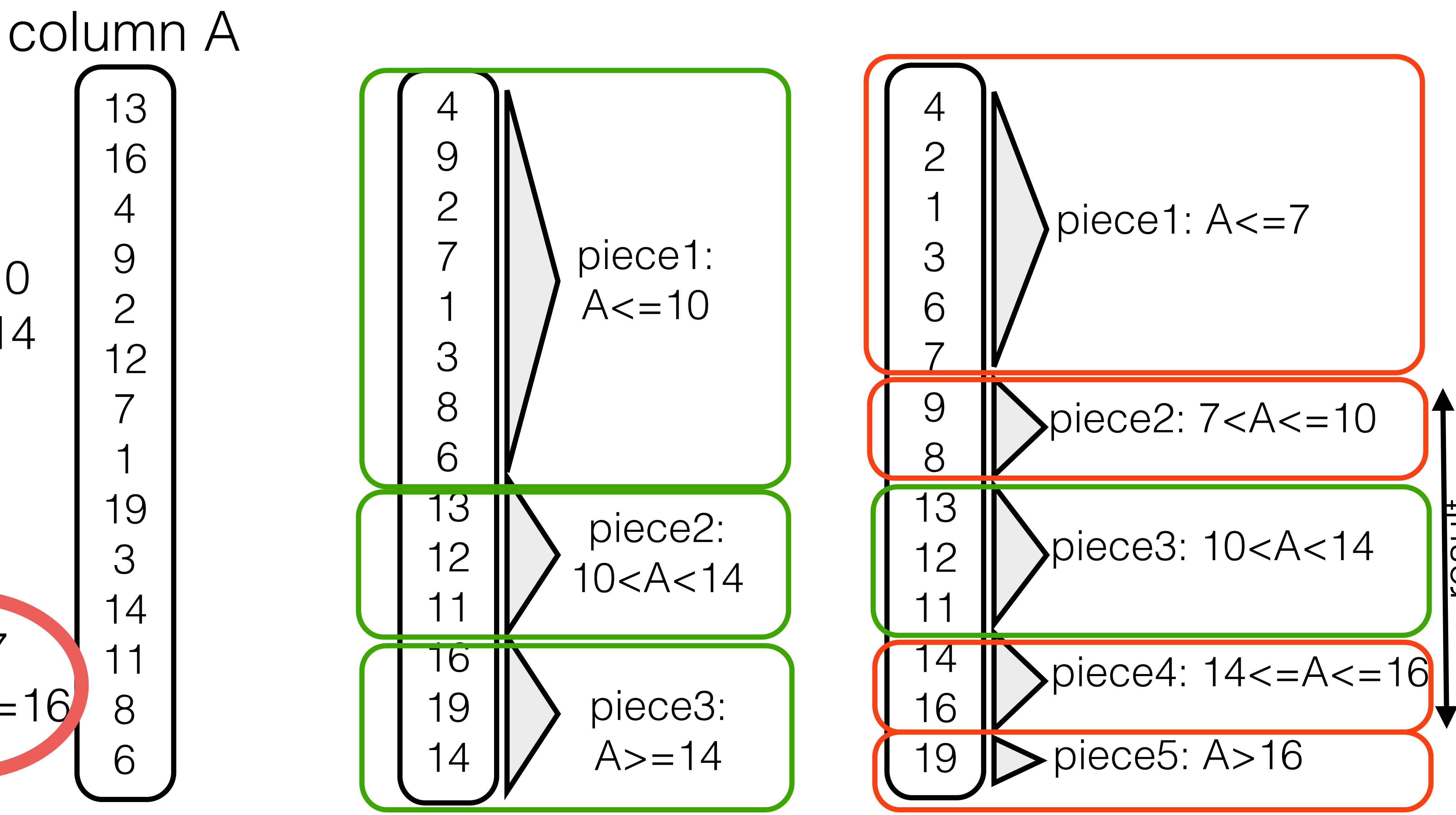
Q2:
select R.A
from R
where R.A>7
and R.A<=16



dynamically/on-the-fly within the select-operator

Q1:
select R.A
from R
where R.A>10
and R.A<14

Q2:
select R.A
from R
where R.A>7
and R.A<=16

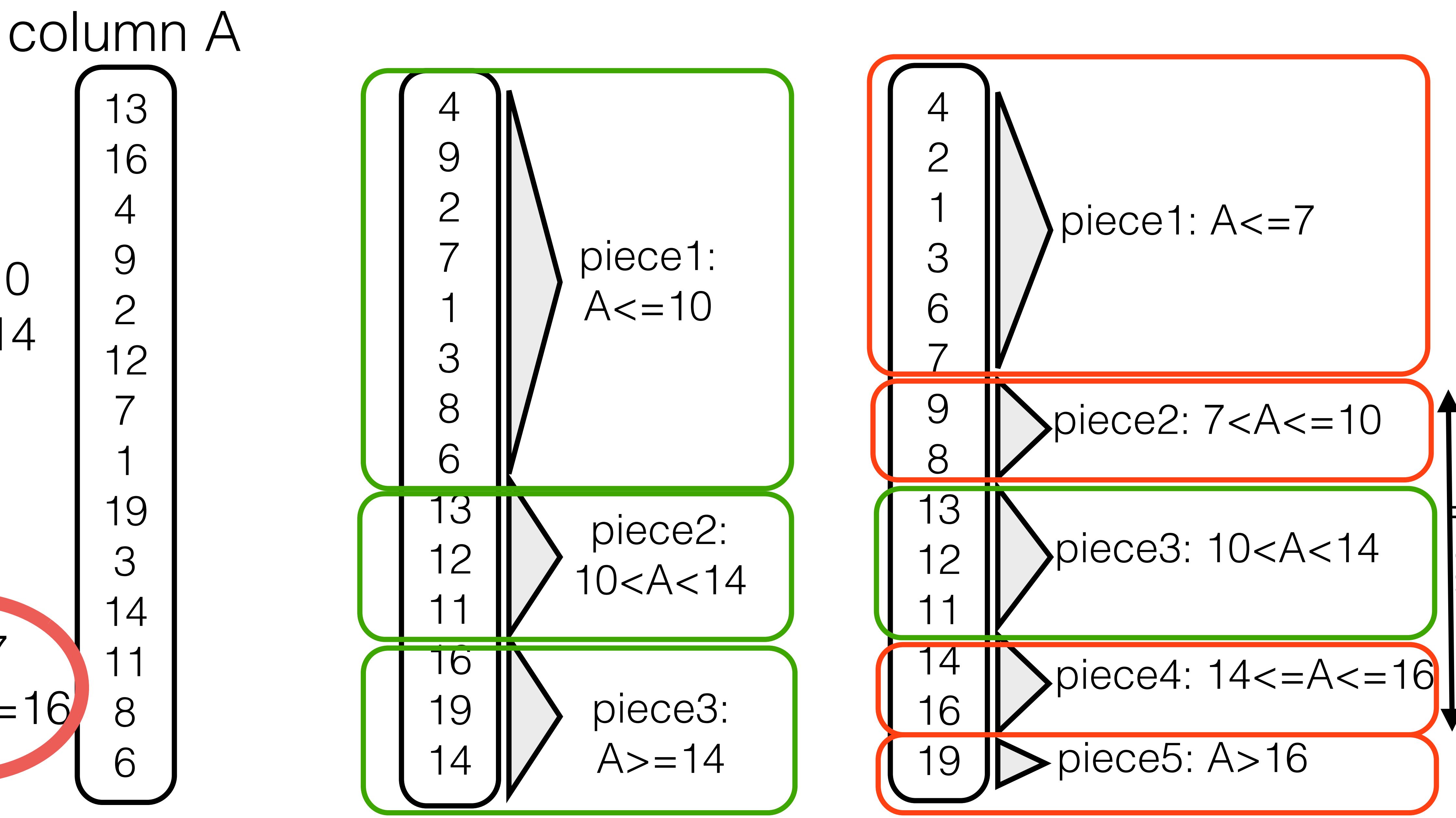


dynamically/on-the-fly within the select-operator

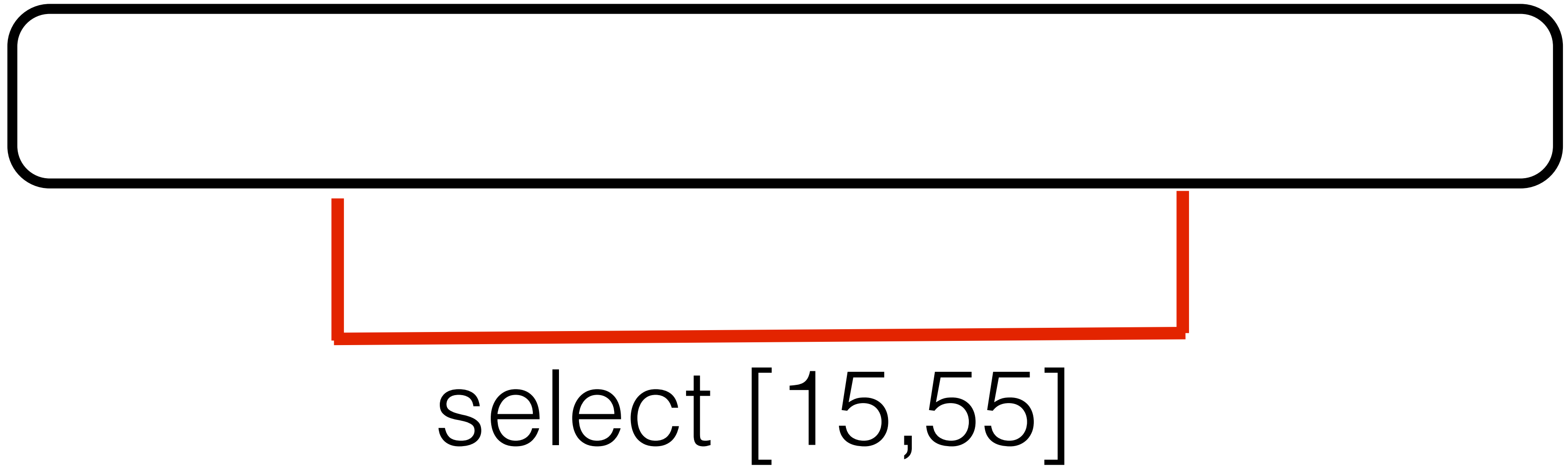
the more we crack, the more we learn

Q1:
select R.A
from R
where R.A>10
and R.A<14

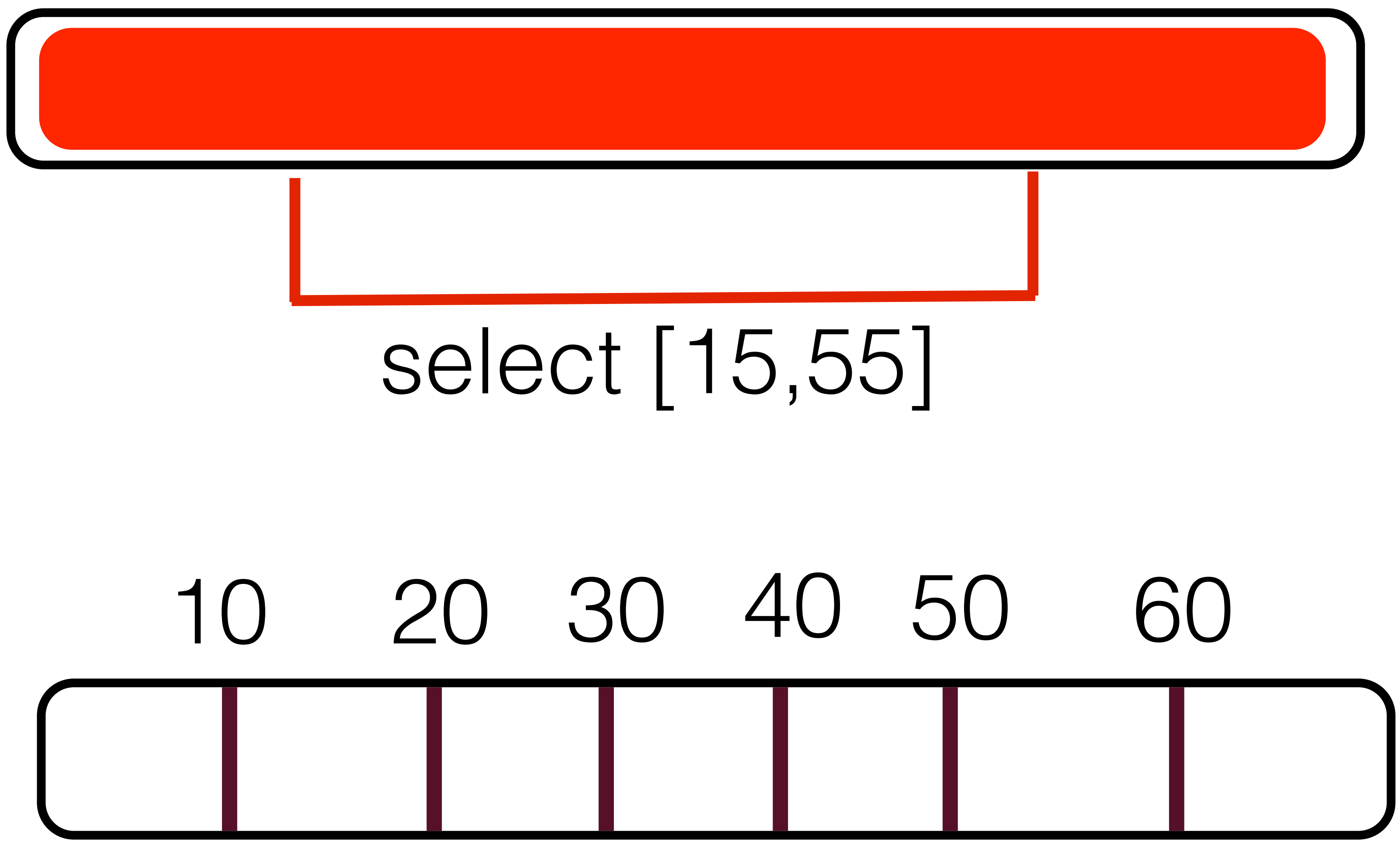
Q2:
select R.A
from R
where R.A>7
and R.A<=16

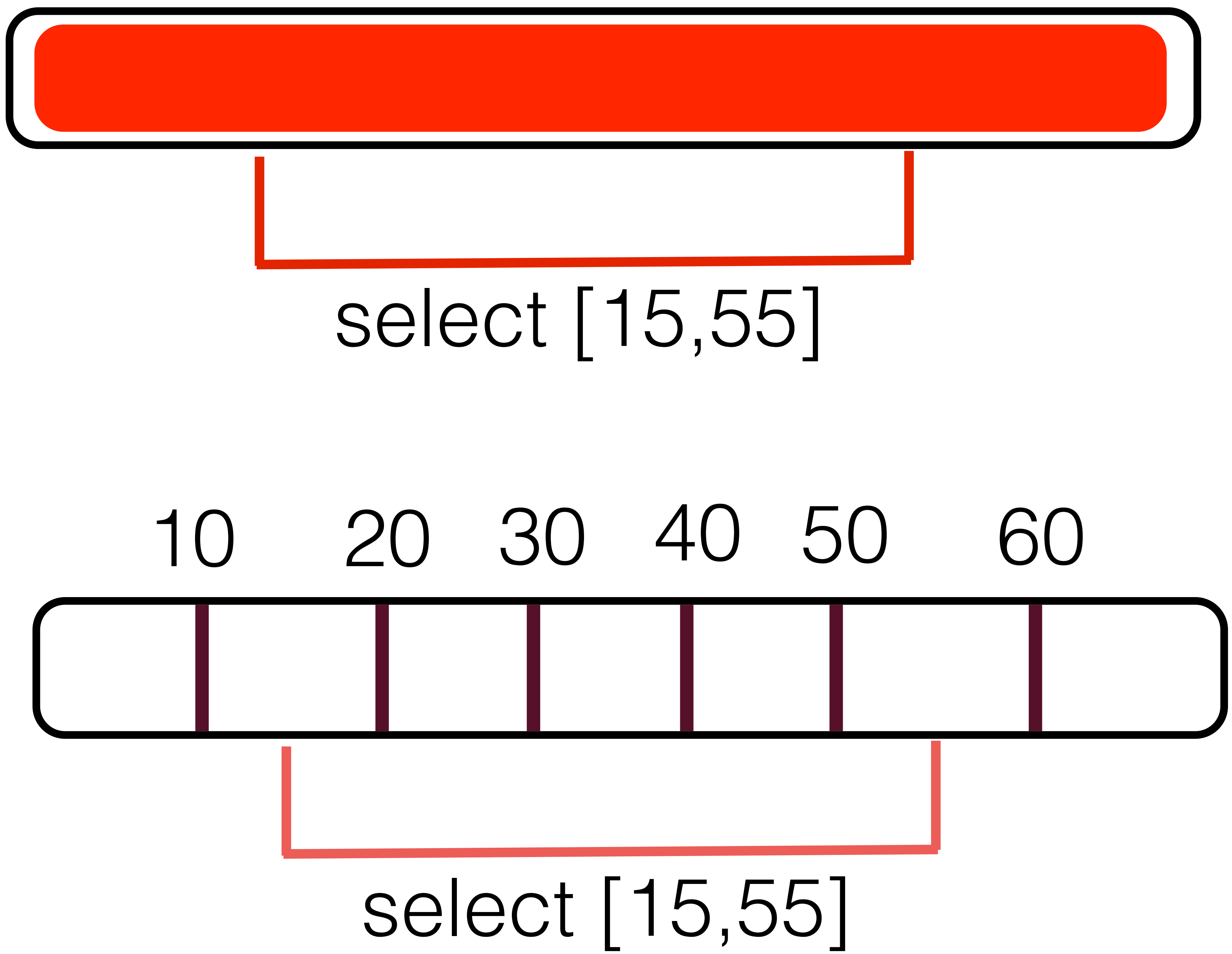


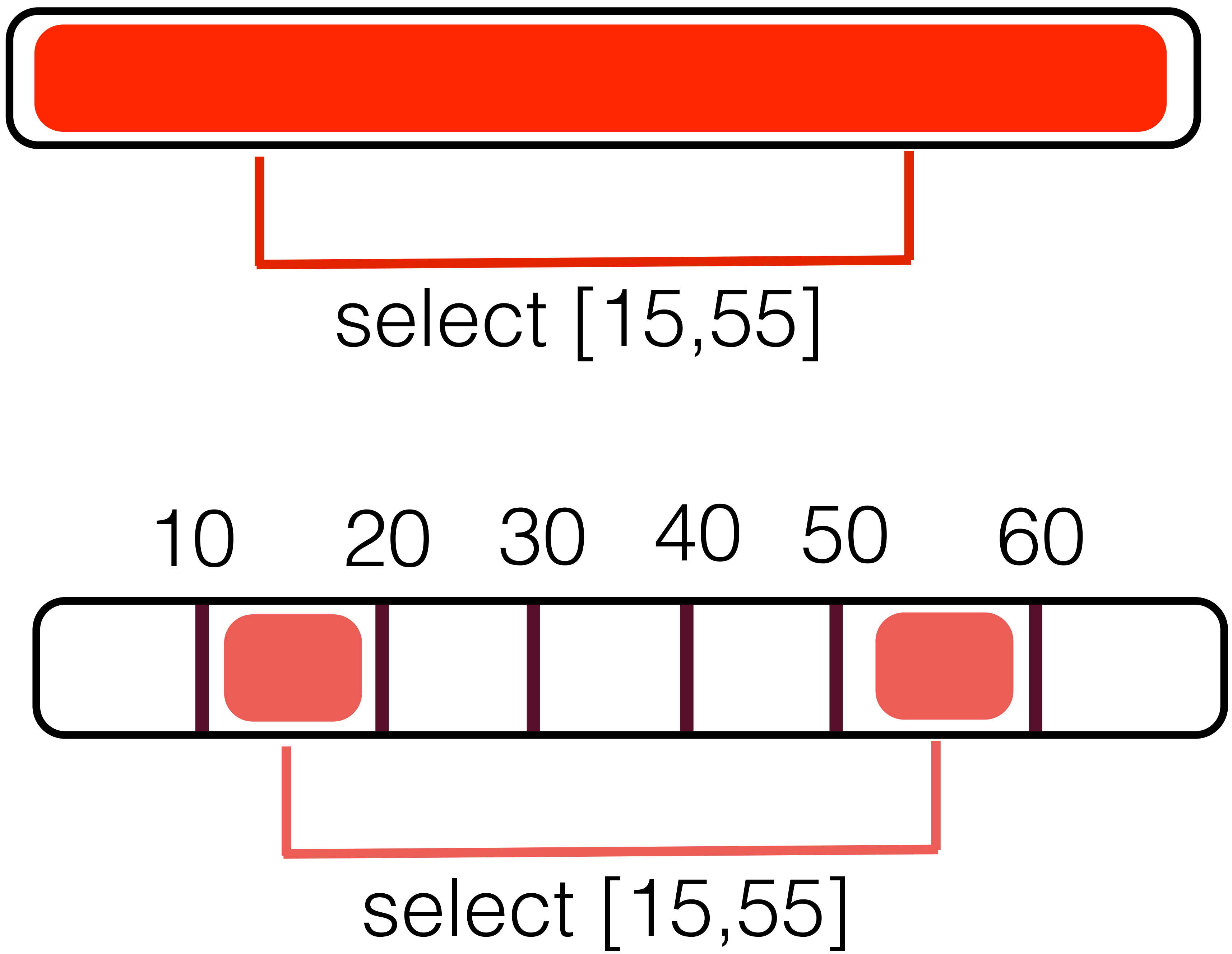
dynamically/on-the-fly within the select-operator


select [15,55]

select [15,55]

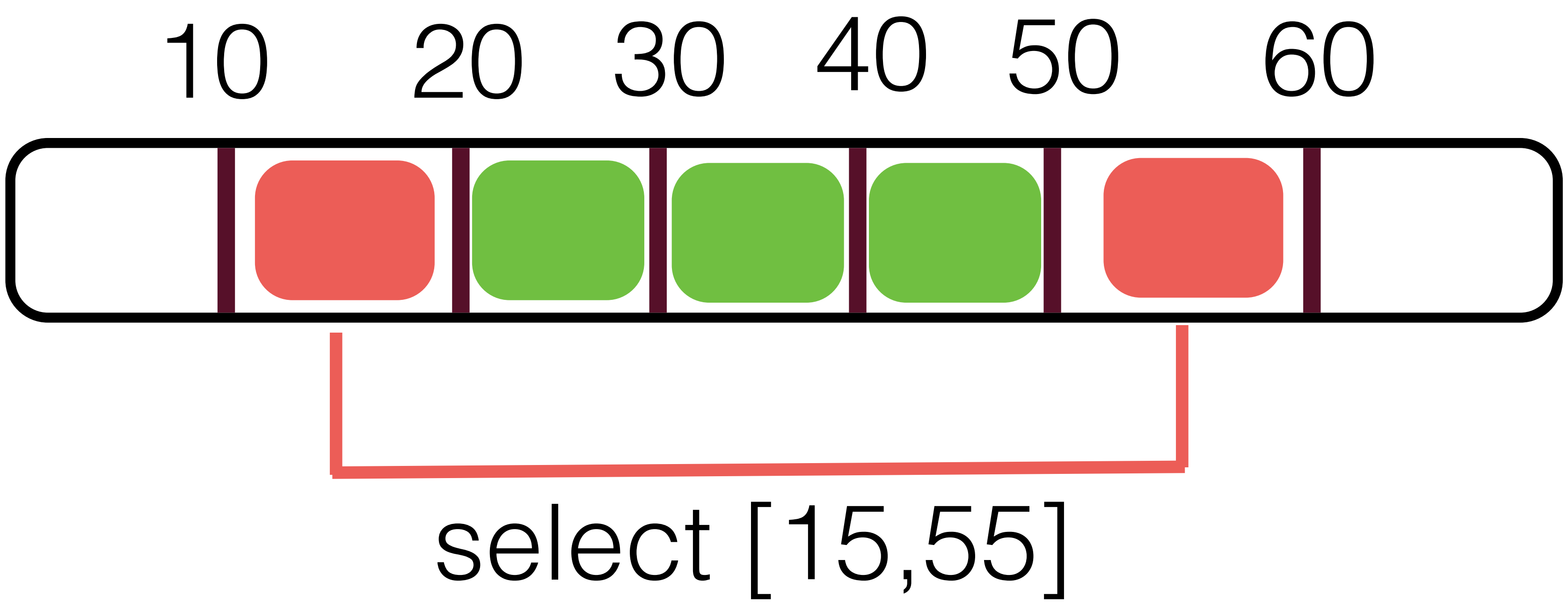

select [15,55]
10
20
30
40
50
60


select [15,55]
10
20
30
40
50
60
select [15,55]


select [15,55]
10
20
30
40
50
60
select [15,55]

touch at most two pieces at a time

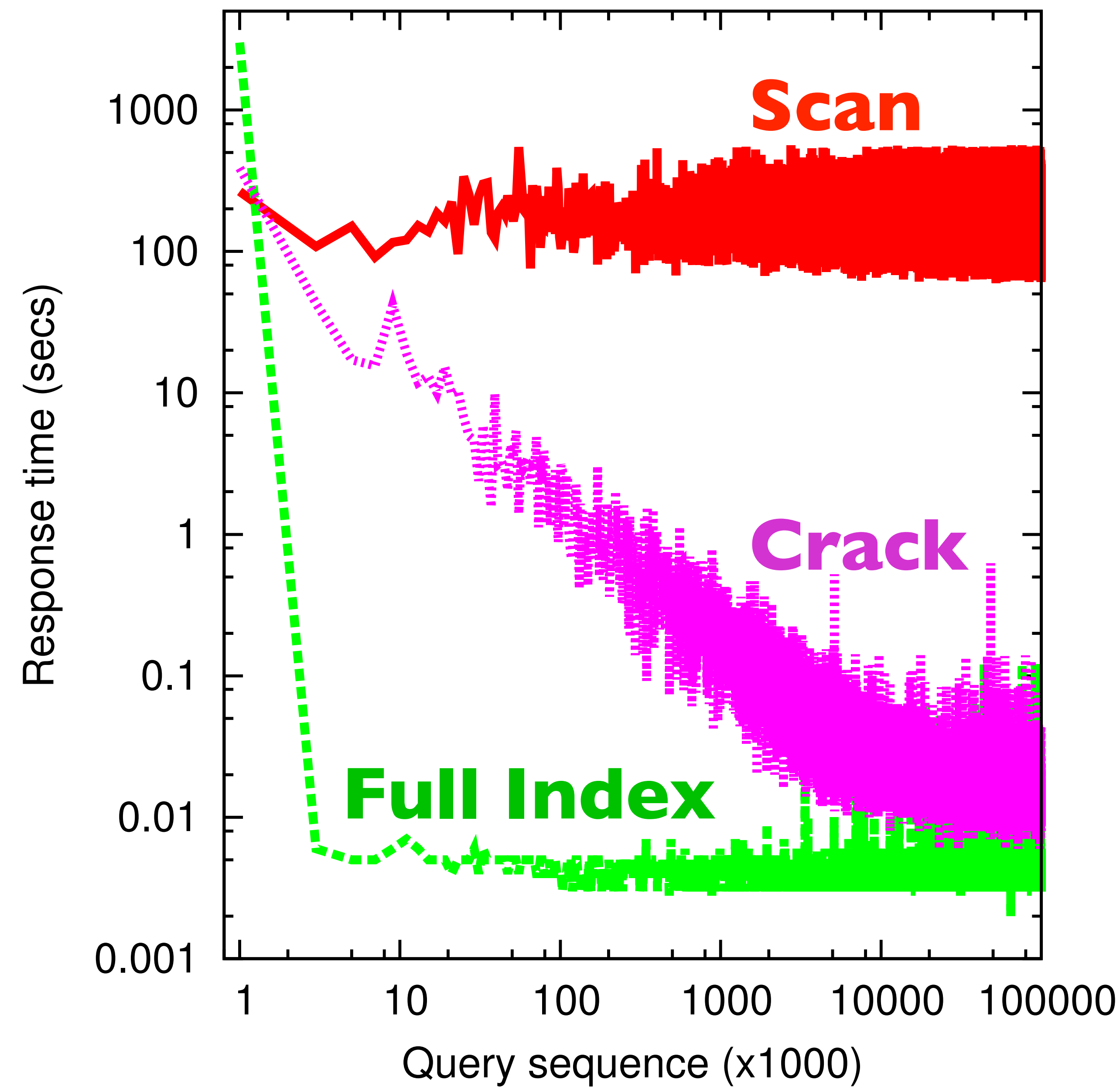
pieces become smaller and smaller



continuous adaptation

set-up

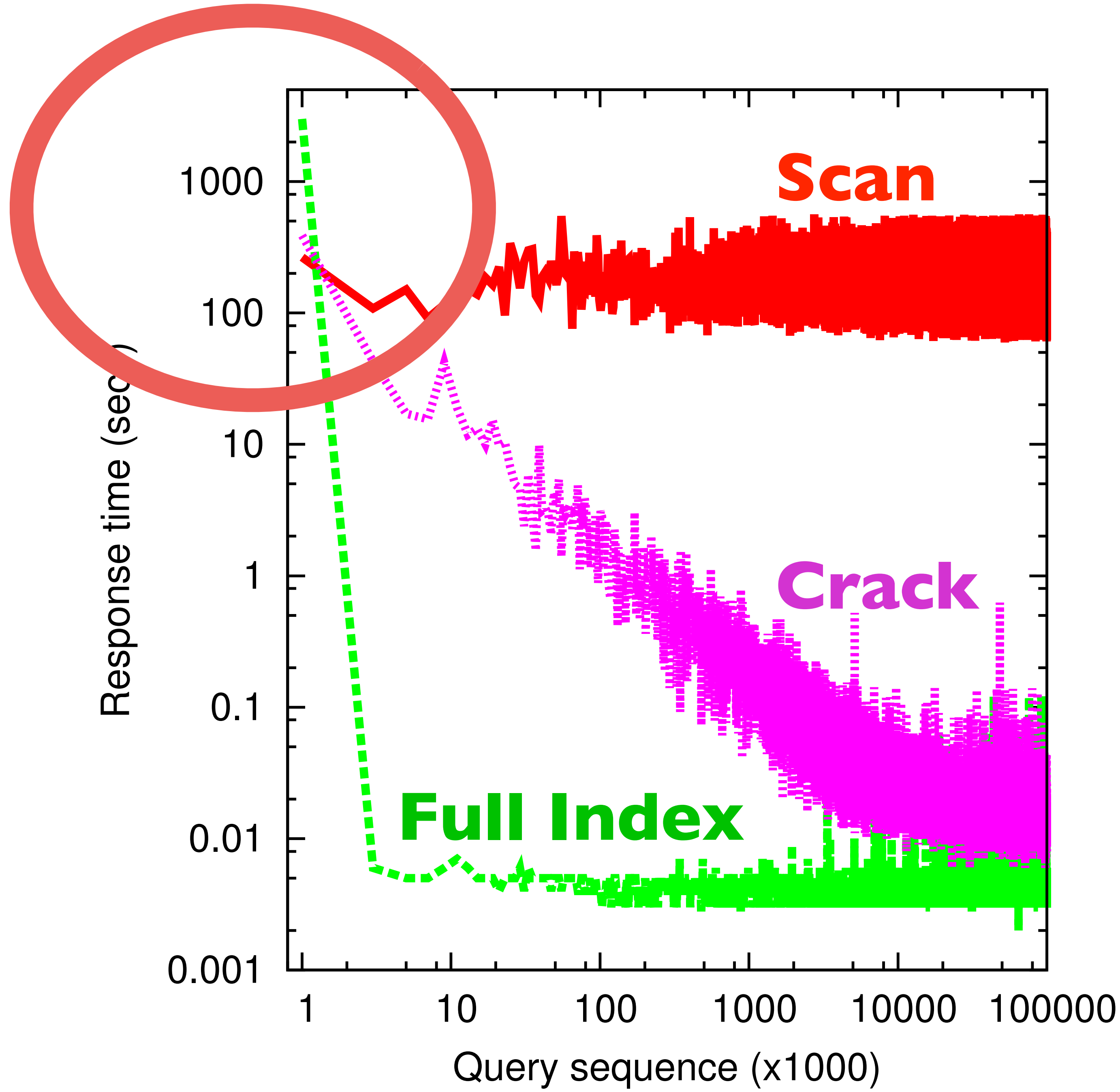
100K random selections
random selectivity
random value ranges
in a 10 million integer column



continuous adaptation

set-up

100K random selections
random selectivity
random value ranges
in a 10 million integer column

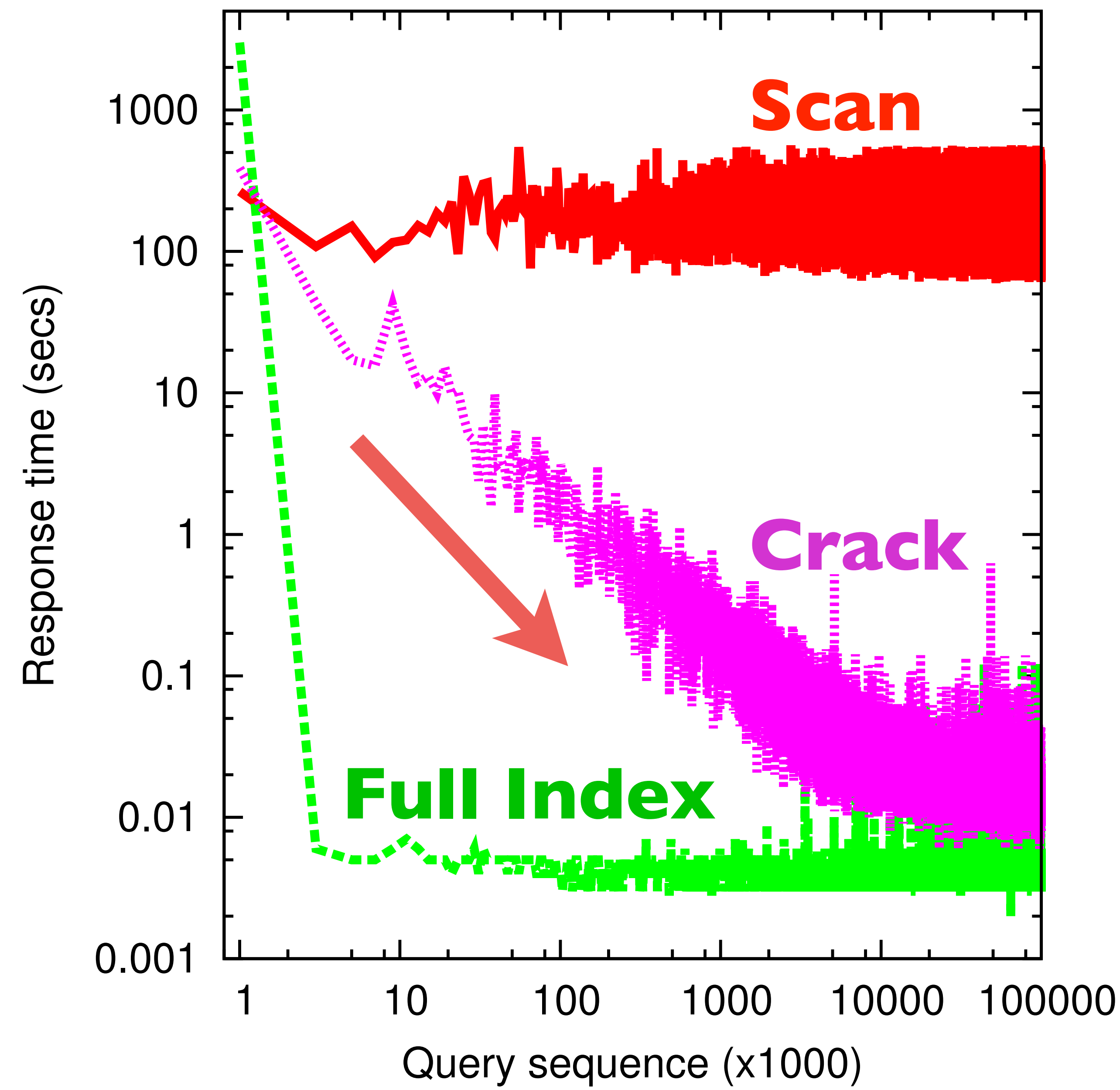


almost no
initialization overhead

continuous adaptation

set-up

100K random selections
random selectivity
random value ranges
in a 10 million integer column



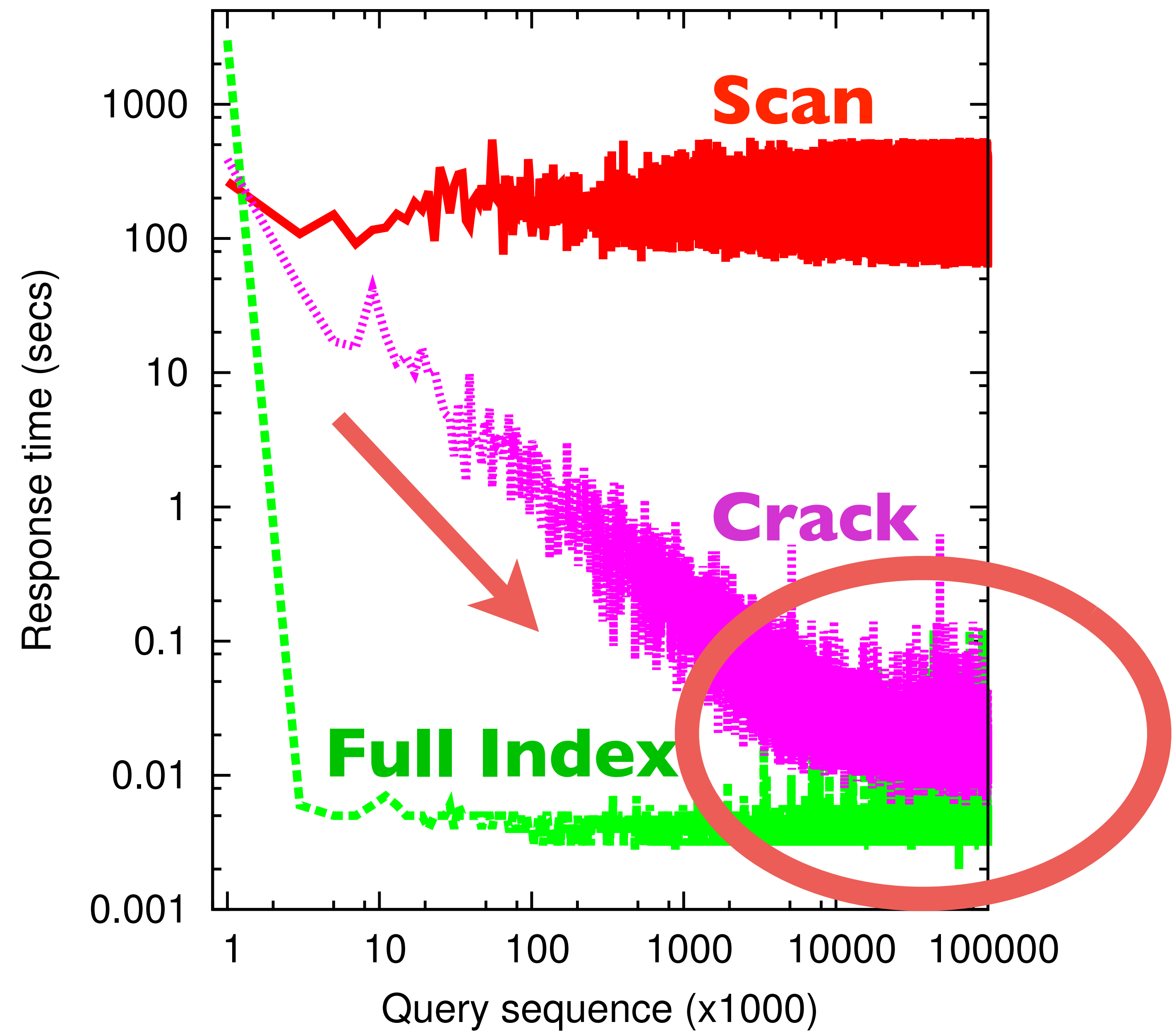
almost no
initialization overhead

continuous improvement

continuous adaptation

set-up

100K random selections
random selectivity
random value ranges
in a 10 million integer column



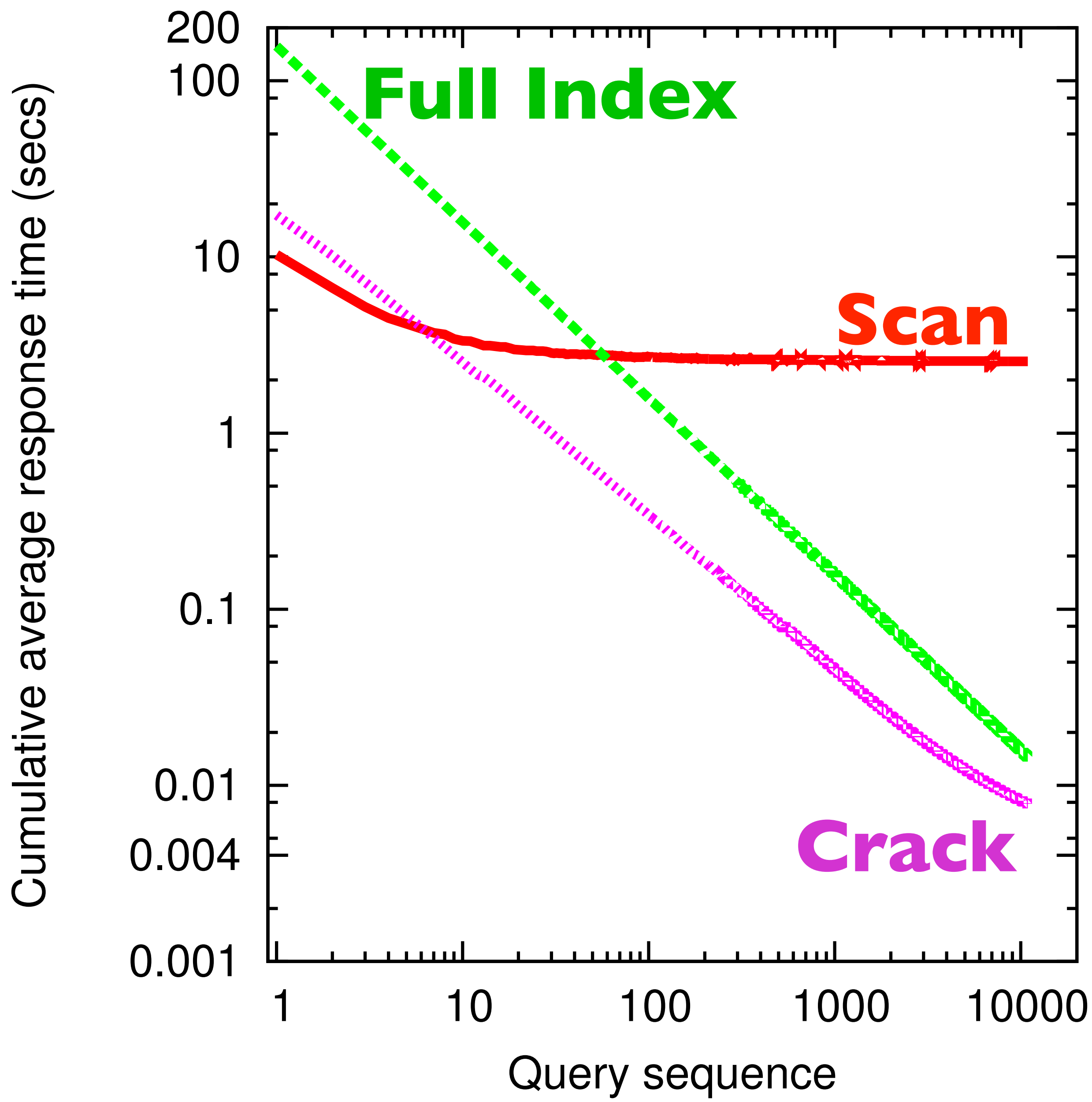
almost no
initialization overhead

continuous improvement

continuous adaptation

set-up

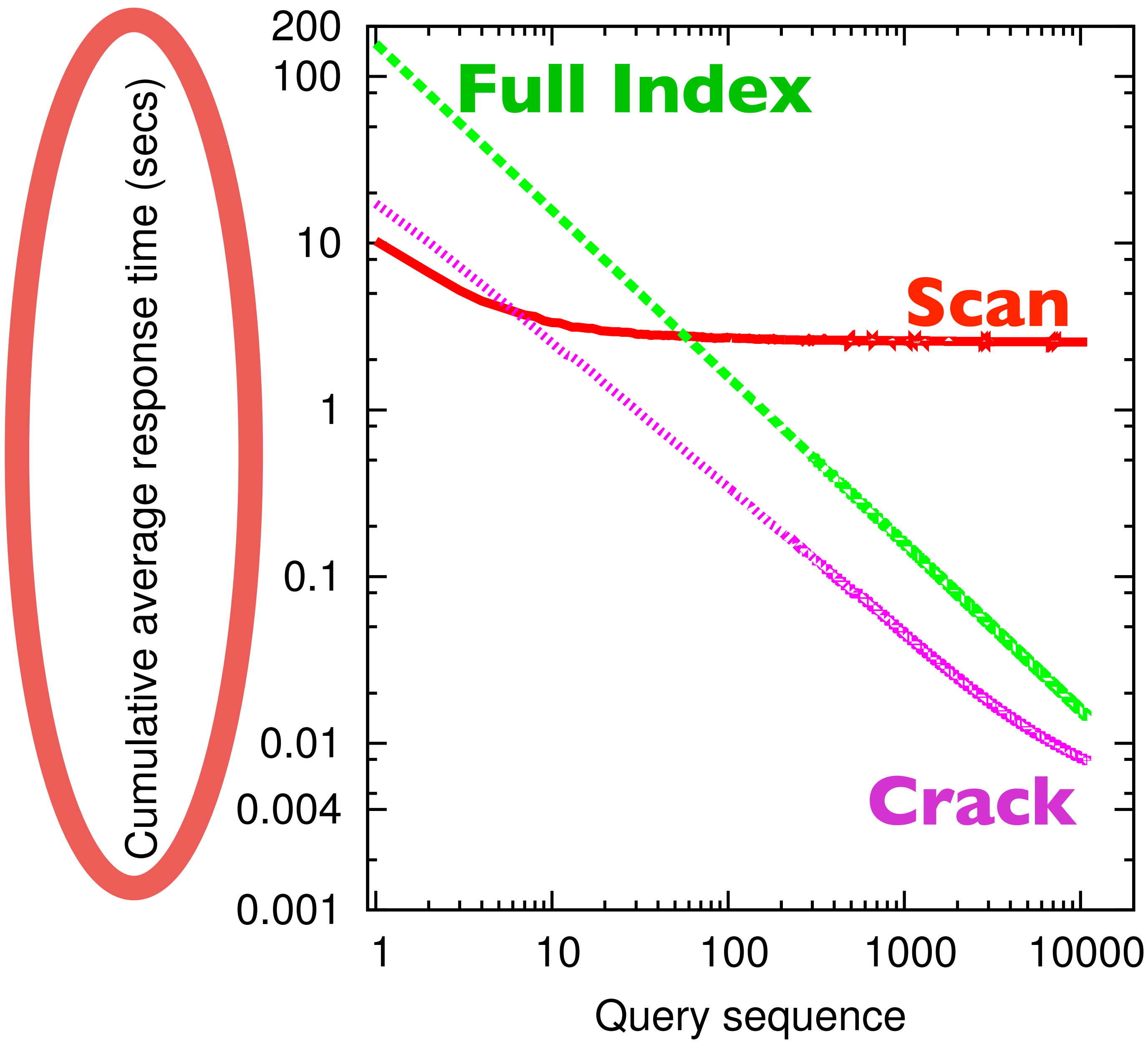
10K random selections
selectivity 10%
random value ranges
in a 30 million integer column



continuous adaptation

set-up

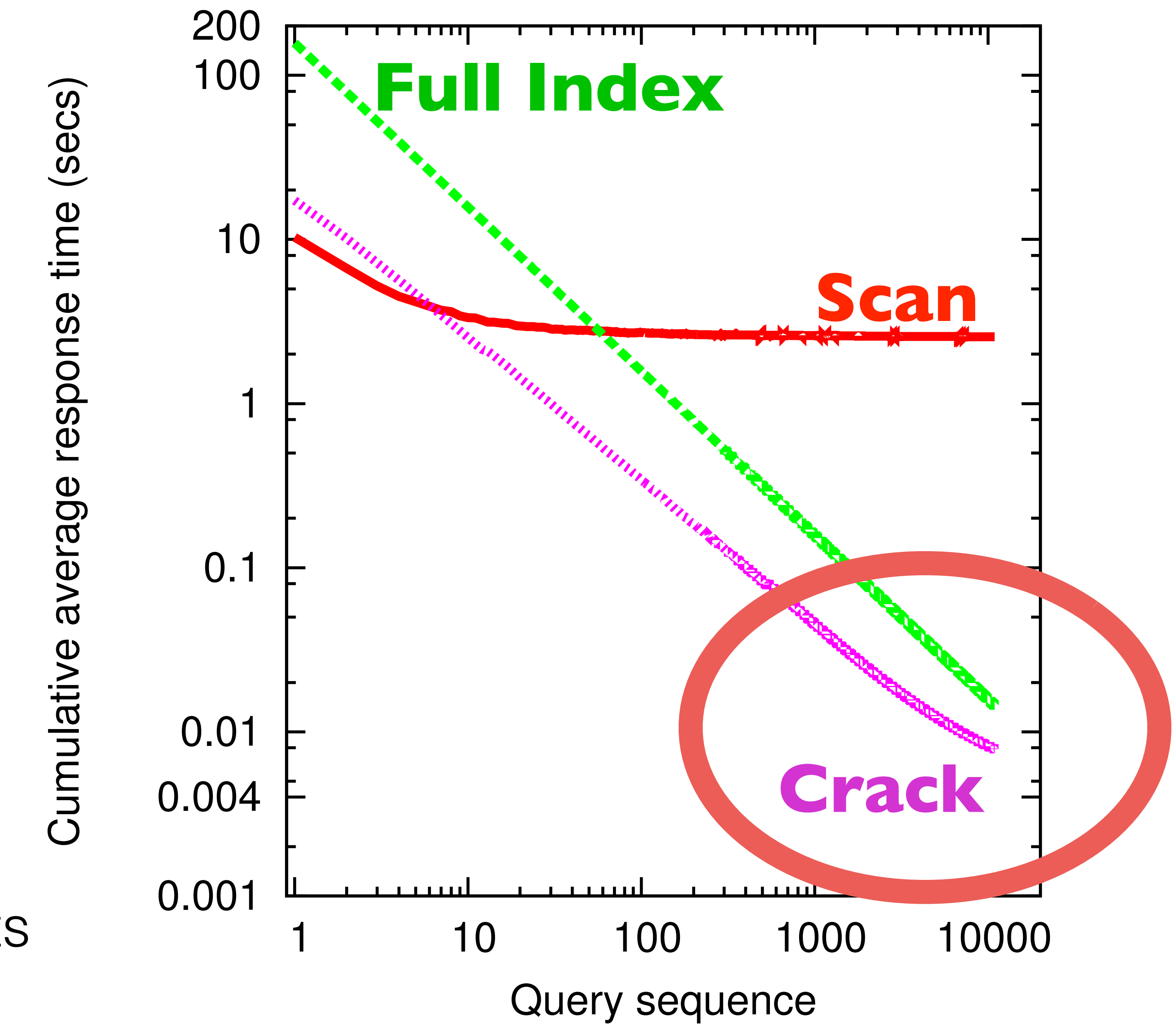
10K random selections
selectivity 10%
random value ranges
in a 30 million integer column



continuous adaptation

set-up

10K random selections
selectivity 10%
random value ranges
in a 30 million integer column



10K queries later,
Full Index still has not
amortized the initialization costs

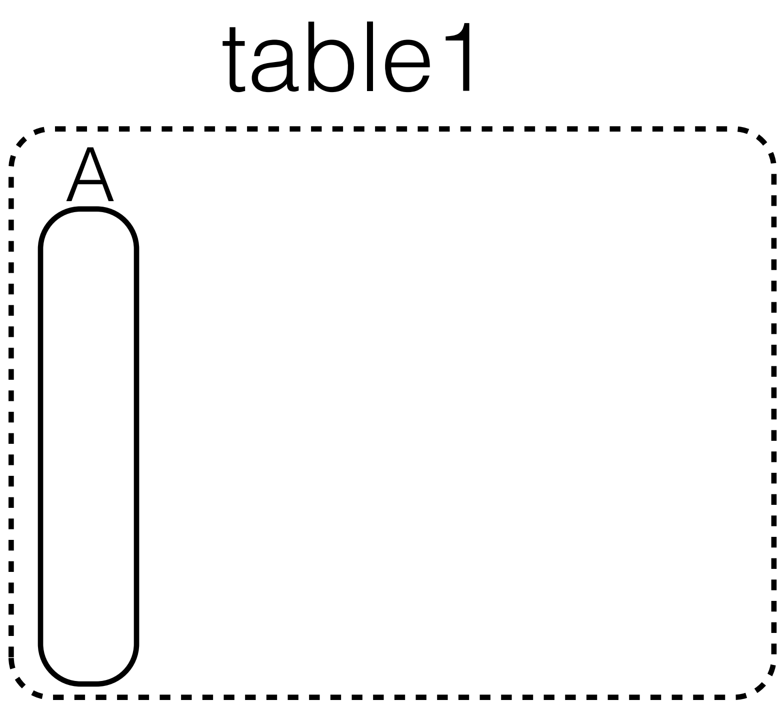


table1
A

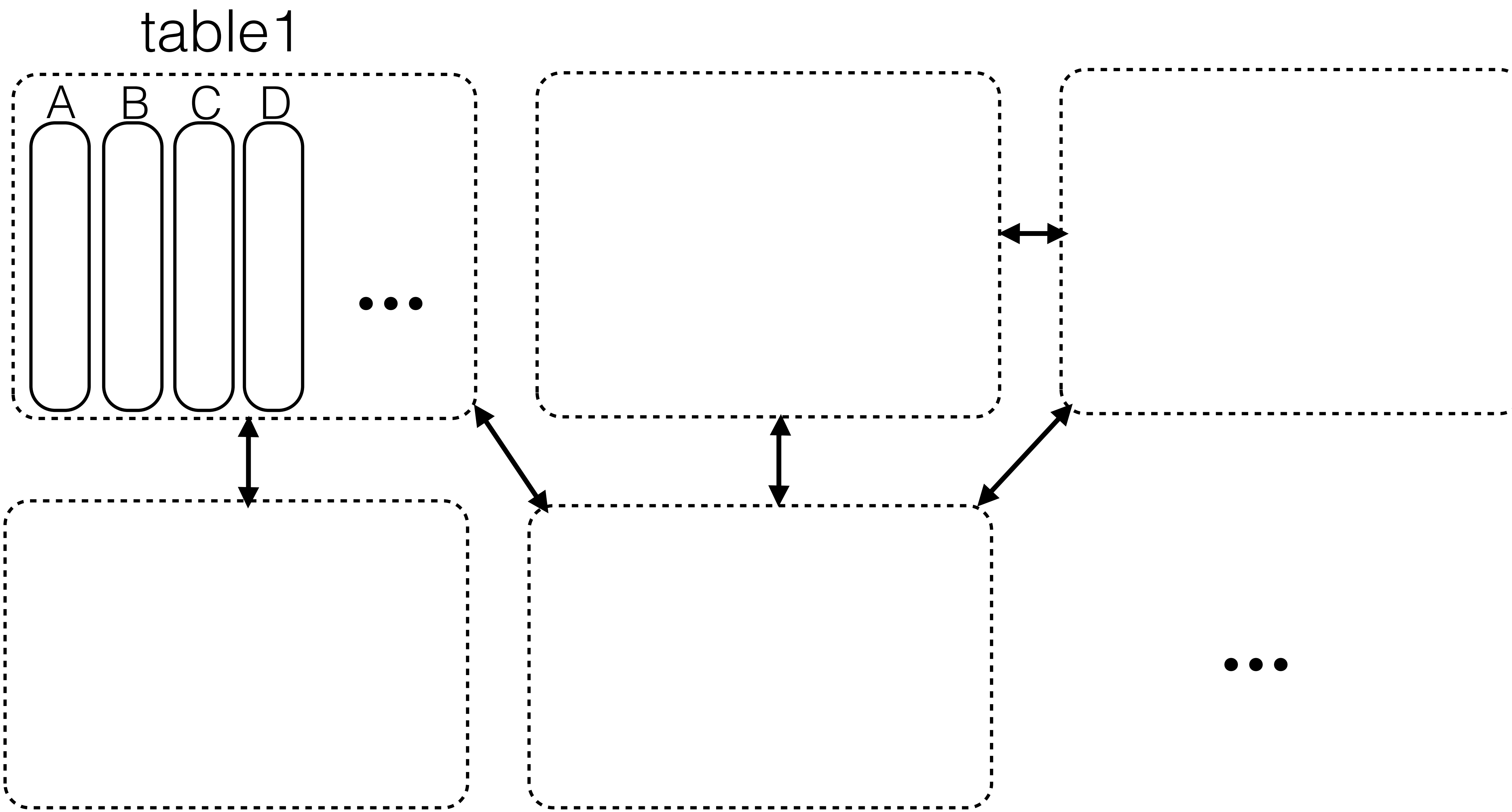
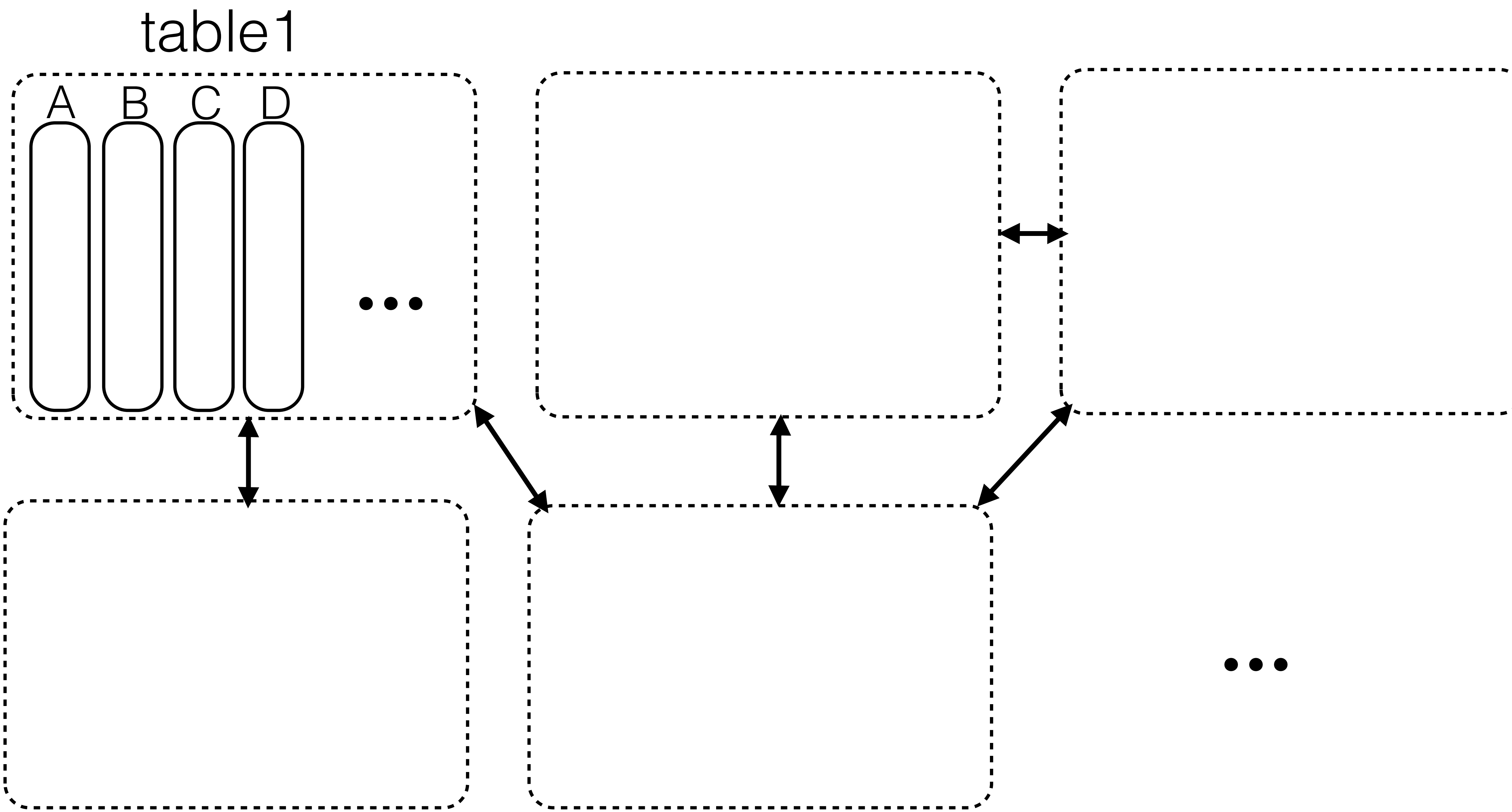


table1
A
B
C
D

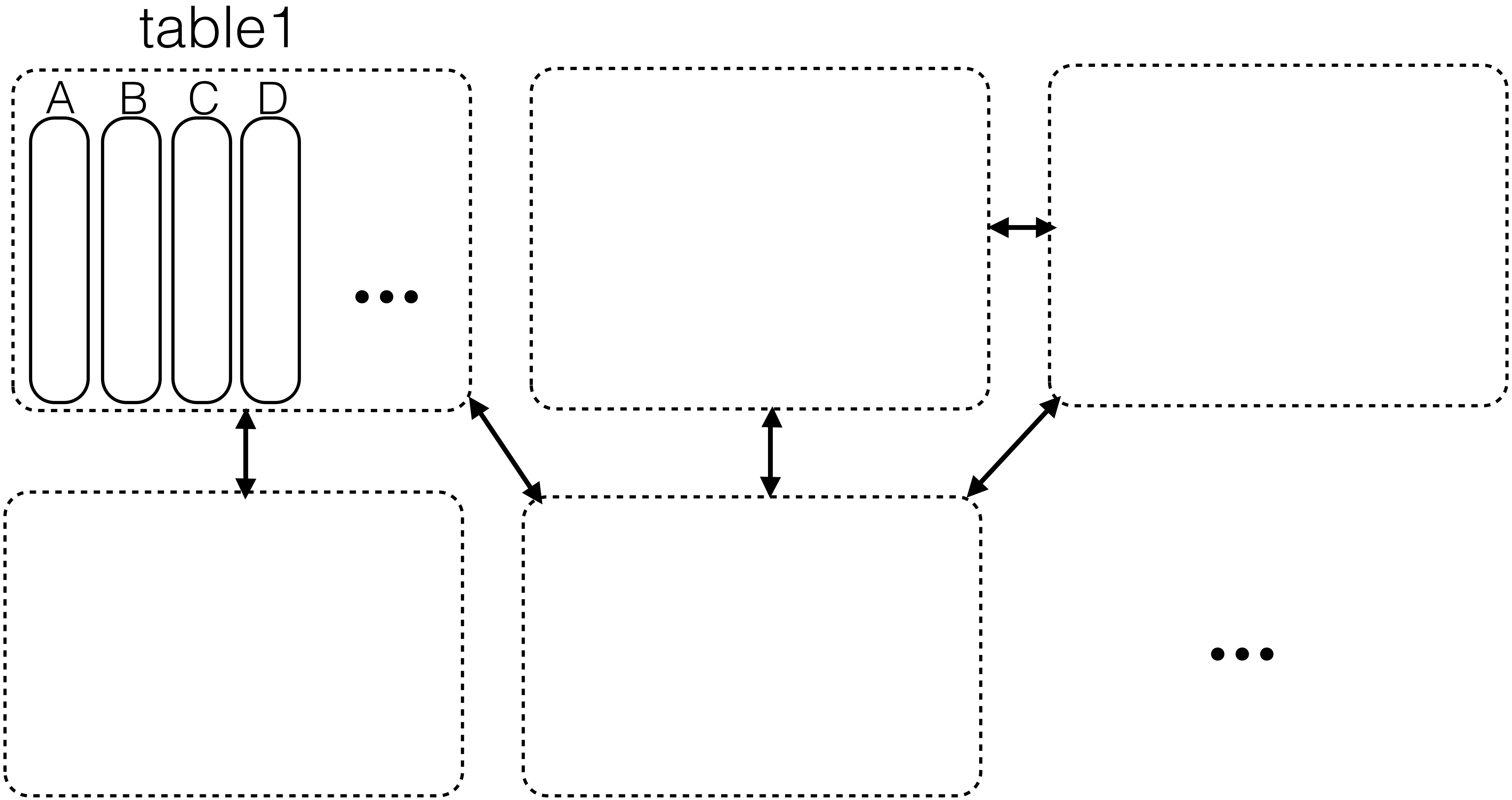
select R.A from R where R.A>10 and R.A<14



select R.A from R where R.A>10 and R.A<14

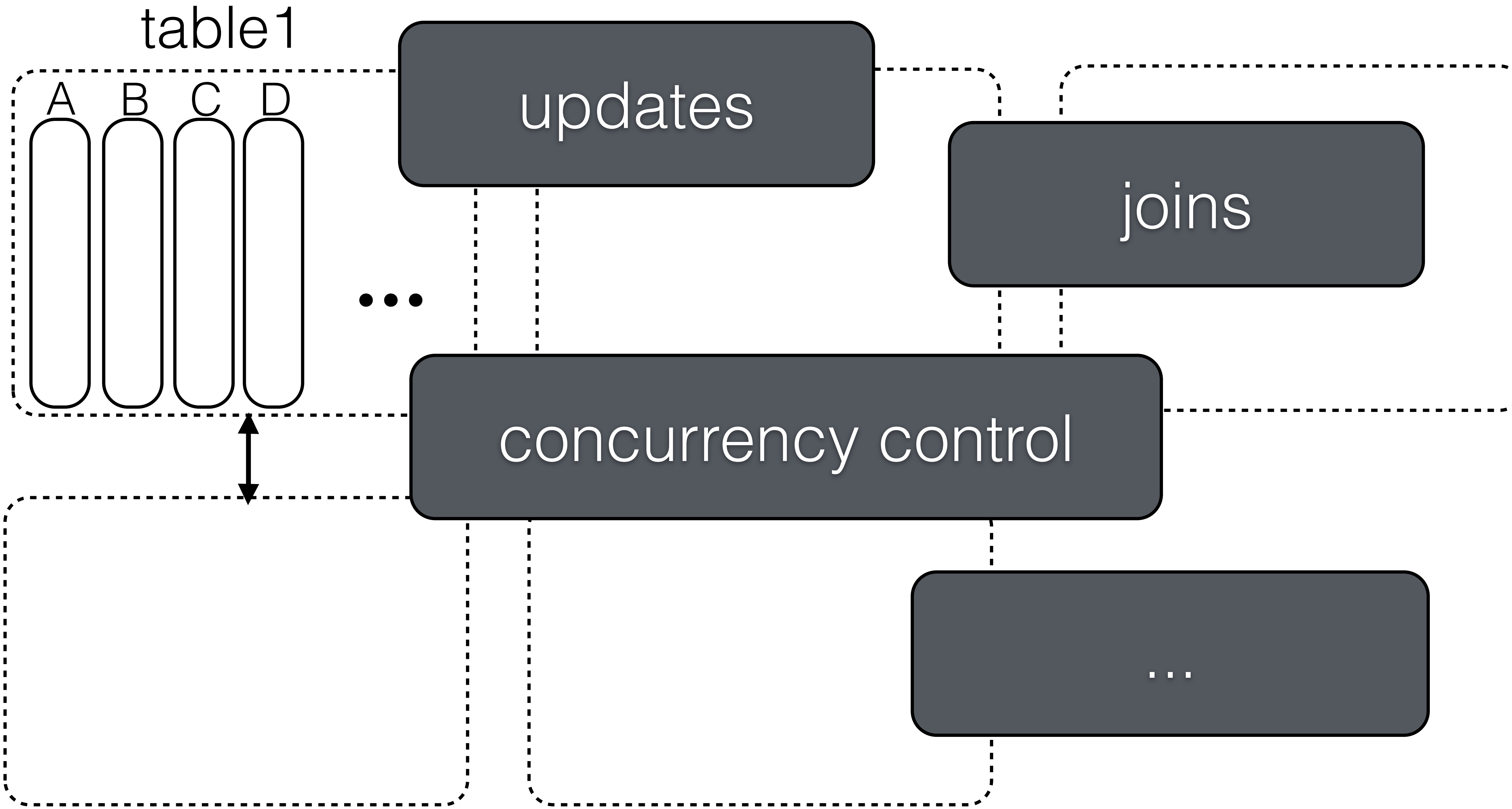
select max(R.A),max(R.B),max(S.A),max(S.B) from R,S
where v1 <R.C<v2 and v3 <R.D<v4
and v5 <R.E<v6 and k1 <S.C<k2 and k3 <S.D<k4 and k5 <S.E<k

and R.F = S.F

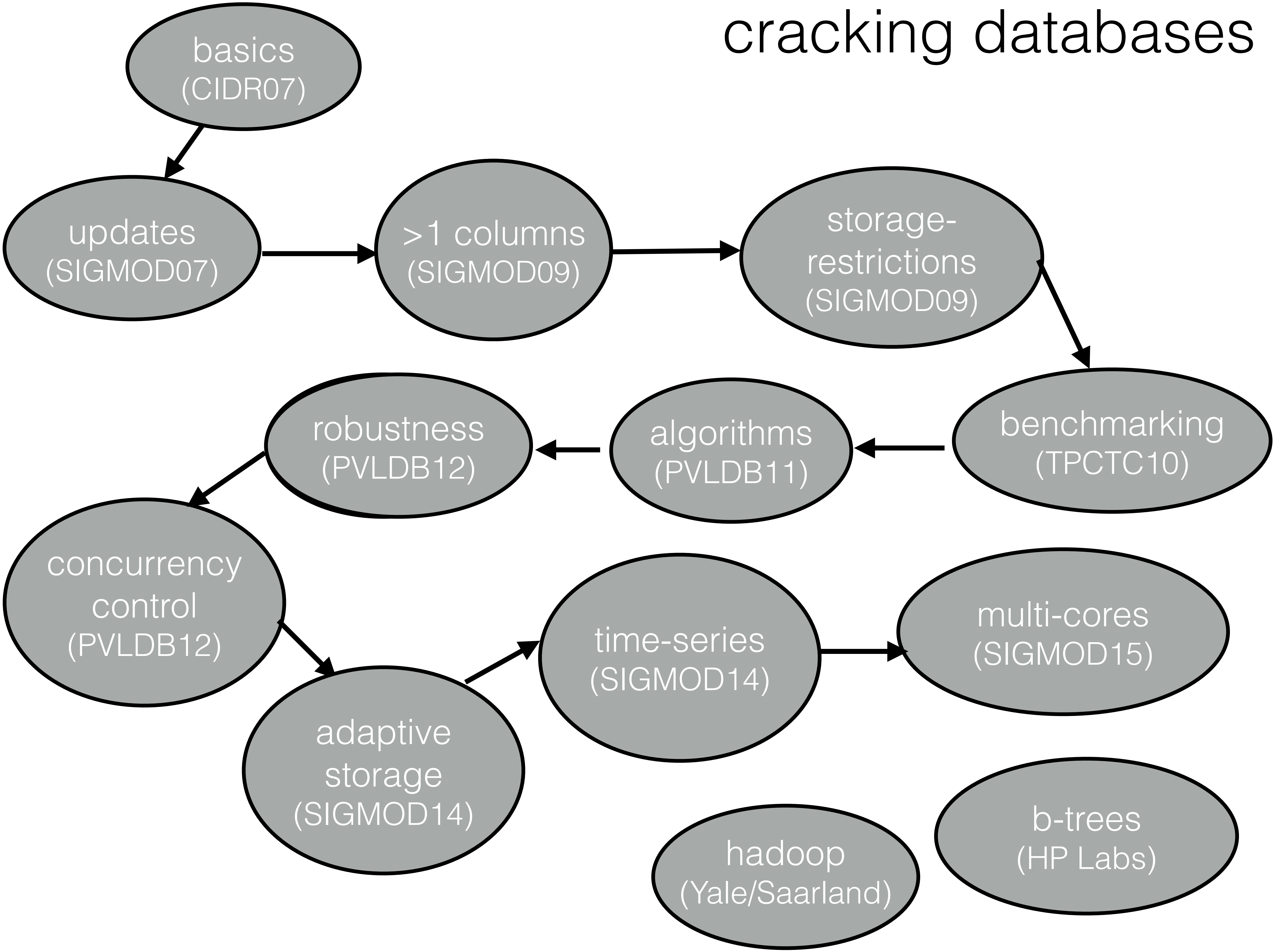


```
select R.A from R where R.A>10 and R.A<14
```

```
select max(R.A),max(R.B),max(S.A),max(S.B) from R,S
where v1 <R.C<v2 and v3 <R.D<v4
and v5 <R.E<v6 and k1 <S.C<k2 and k3 <S.D<k4 and k5 <S.E<k
and R.F = S.F
```



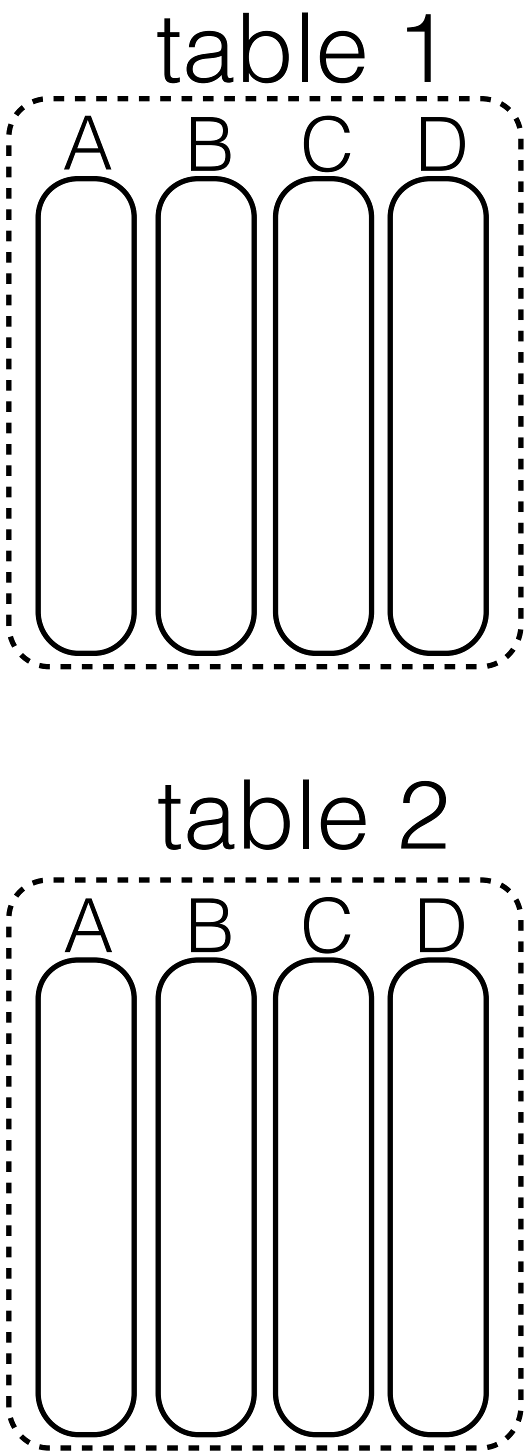
cracking databases



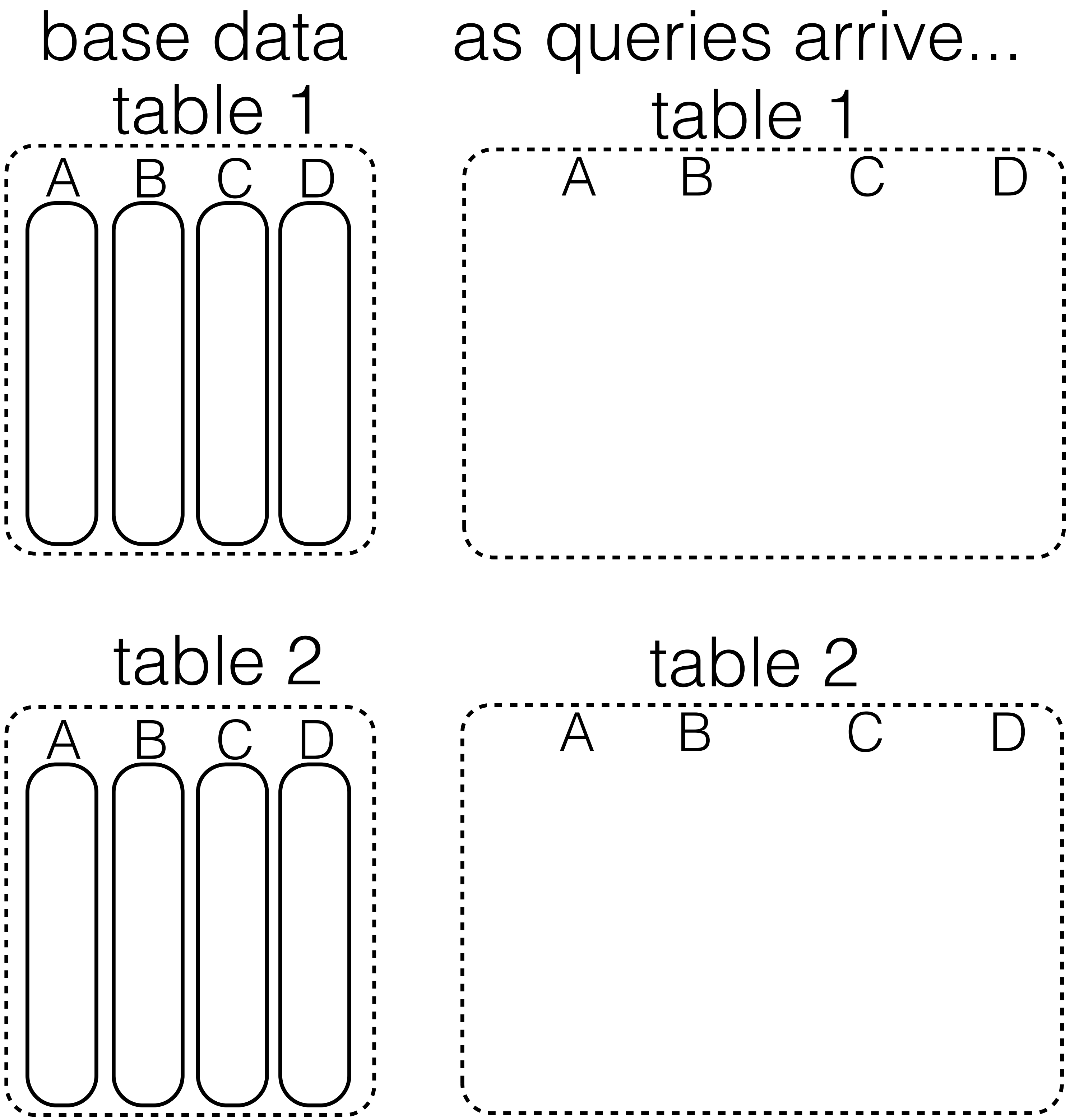
cracking tangram

base data

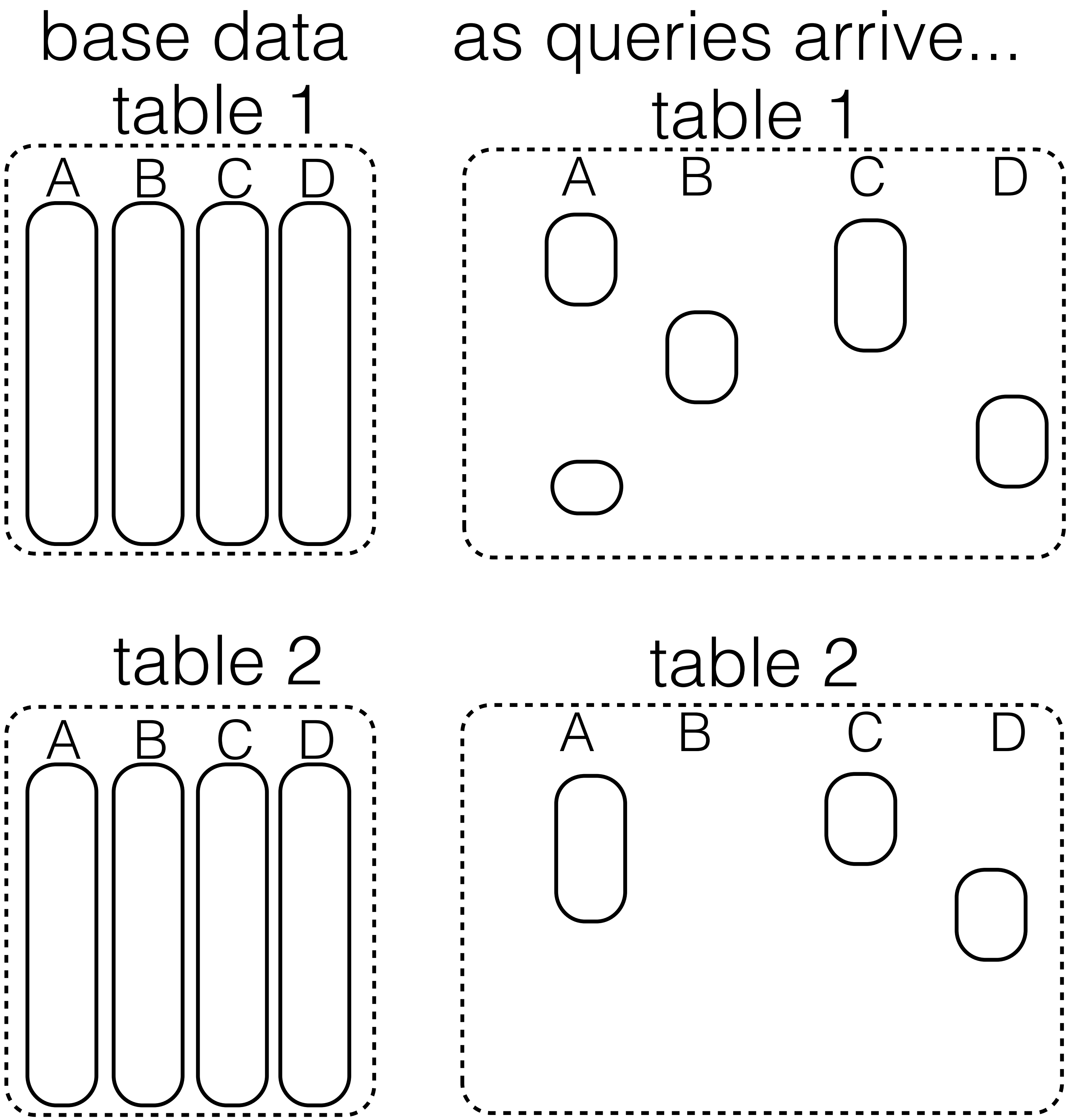
as queries arrive...



cracking tangram



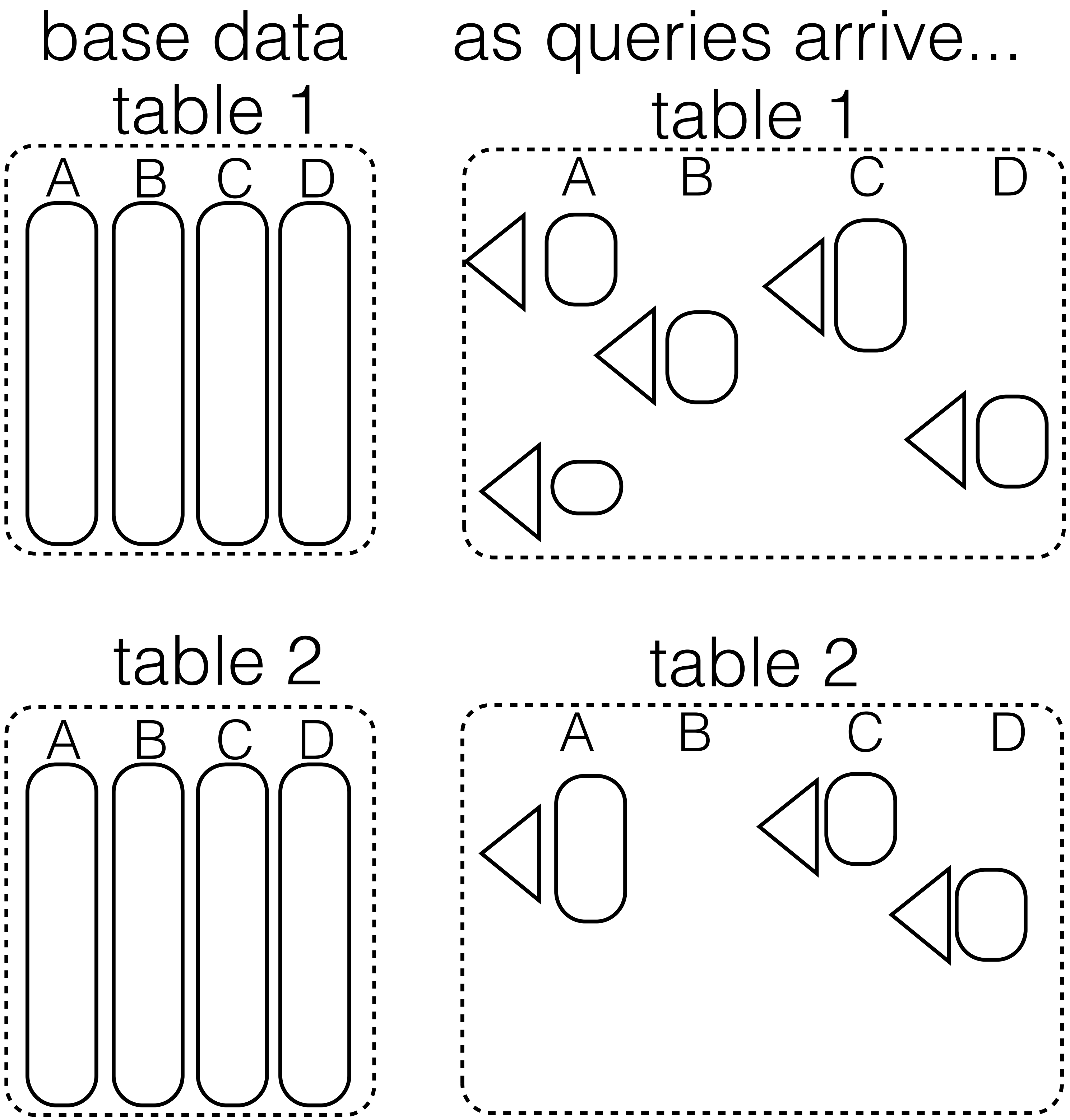
cracking tangram



partial materialization

HARVARD
School of Engineering
and Applied Sciences
Stratos Idreos

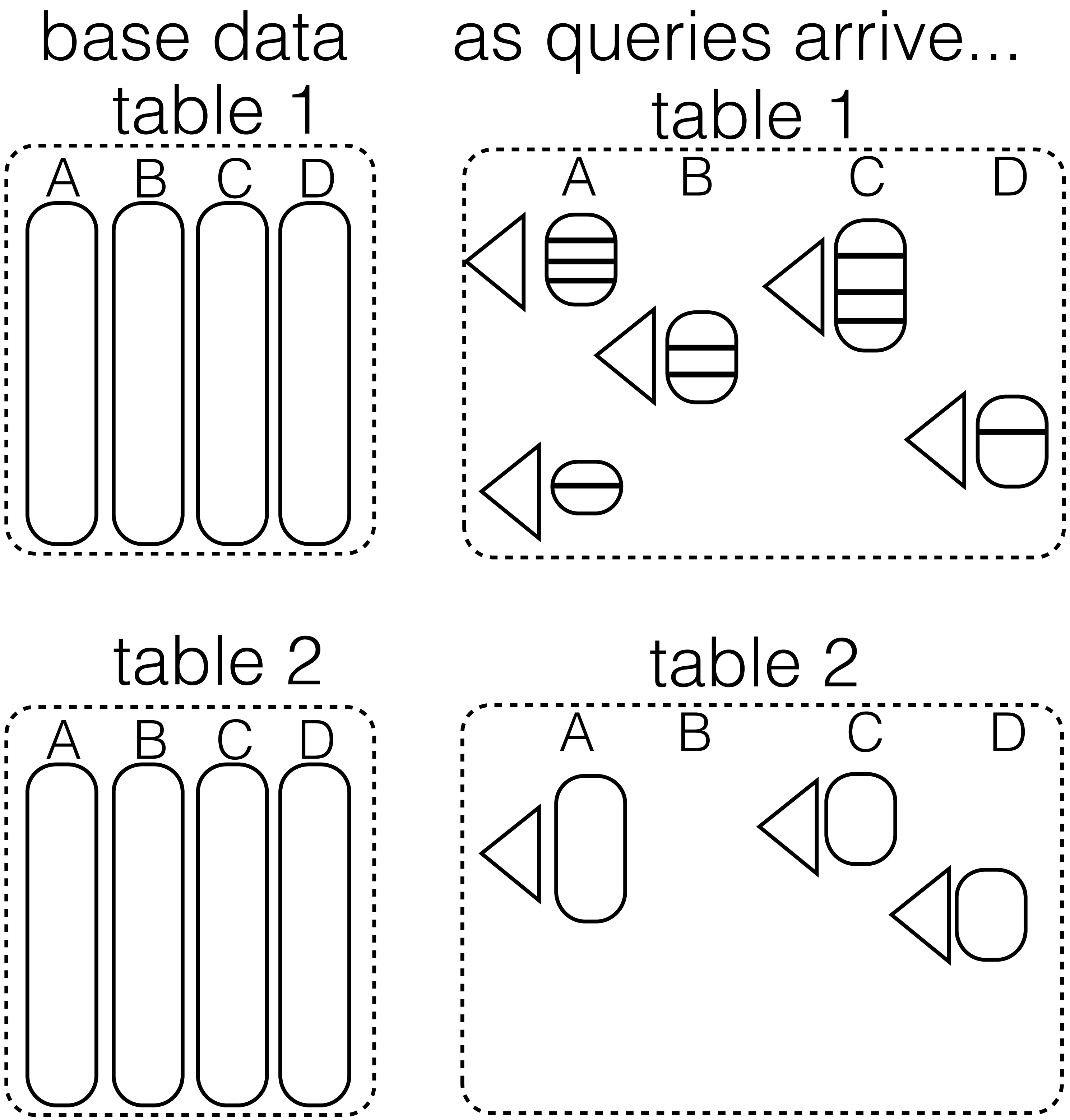
cracking tangram



partial materialization

partial indexing

cracking tangram

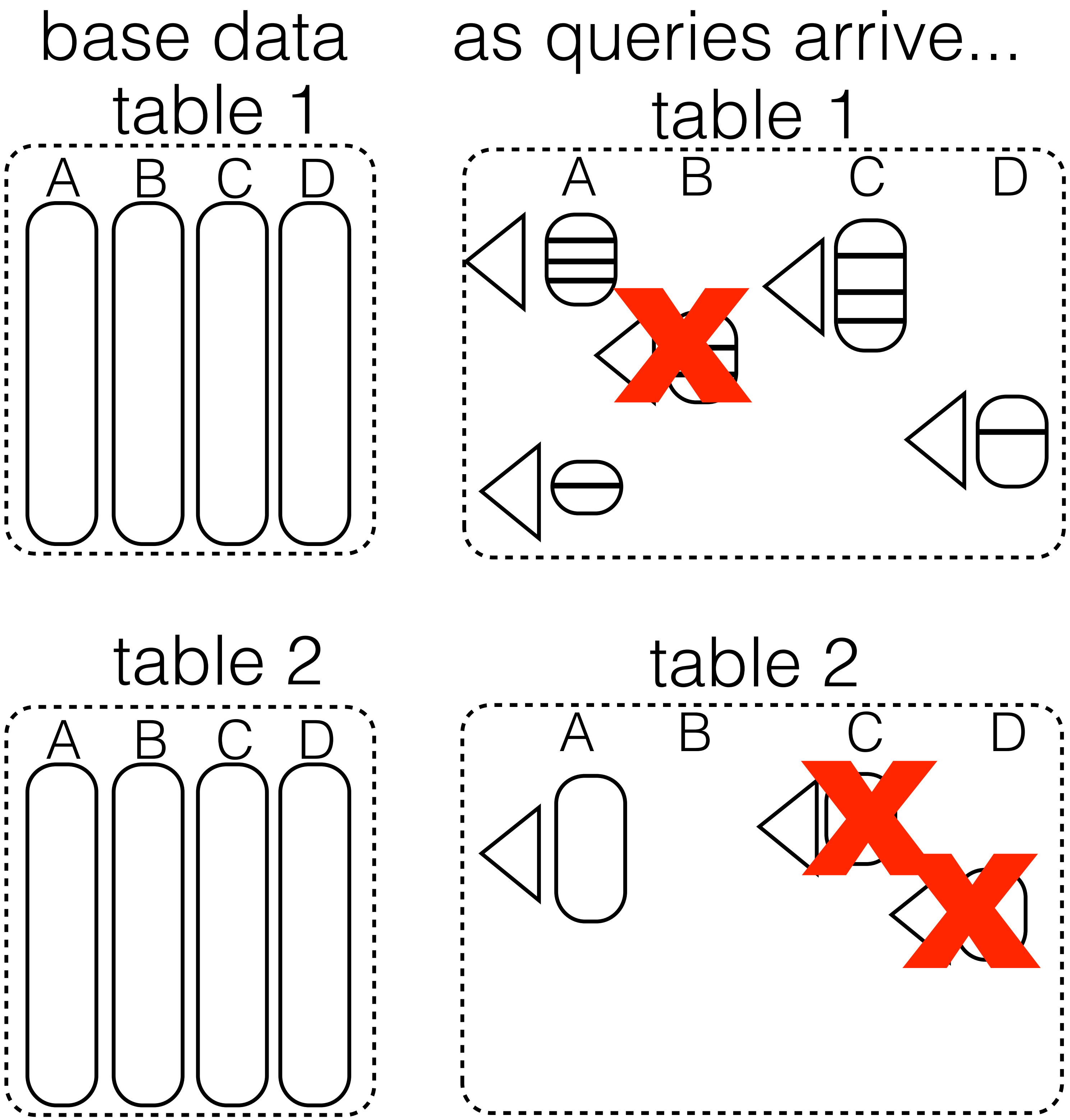


partial materialization

partial indexing

continuous adaptation

cracking tangram



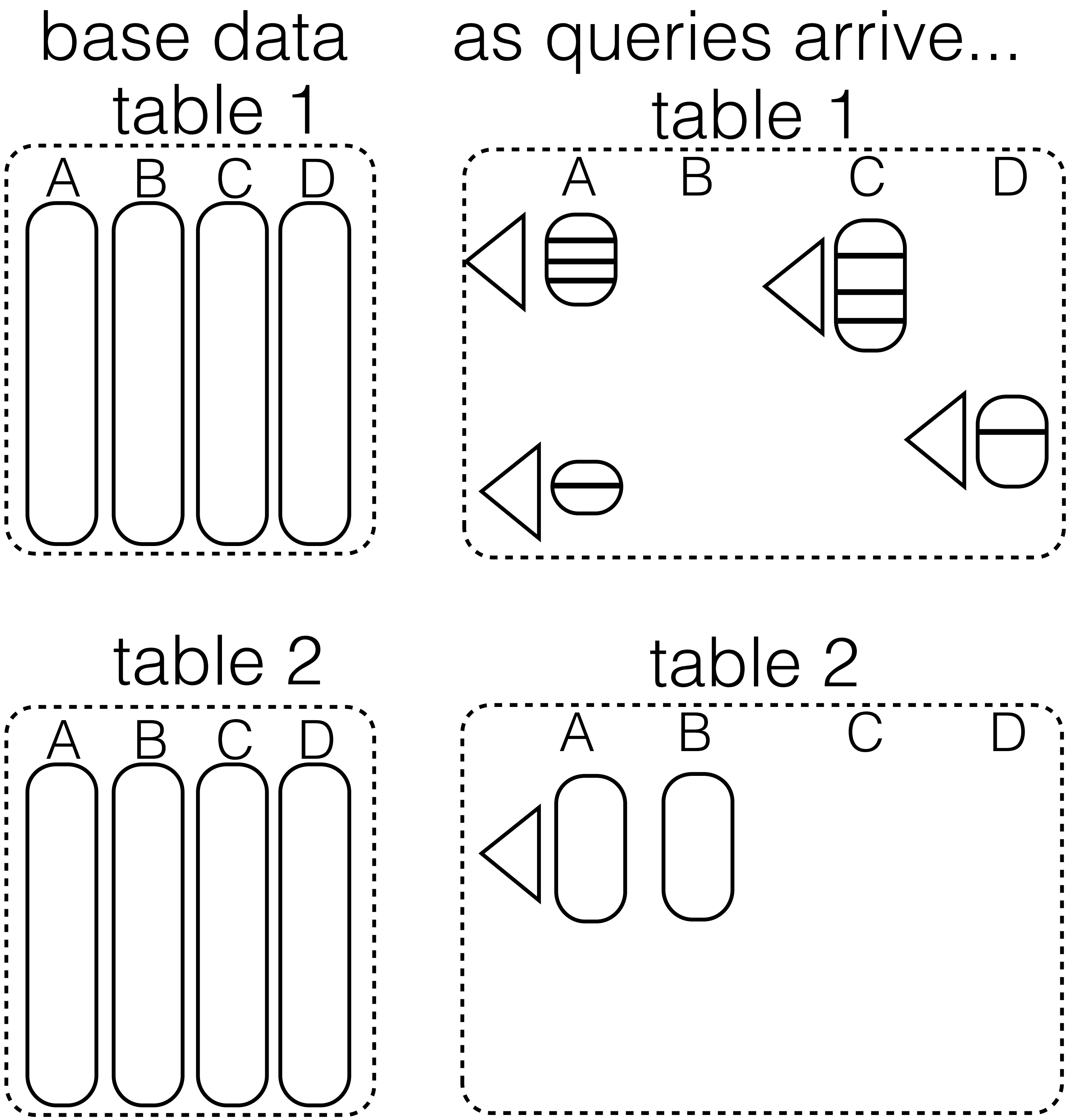
partial materialization

partial indexing

continuous adaptation

storage adaptation

cracking tangram



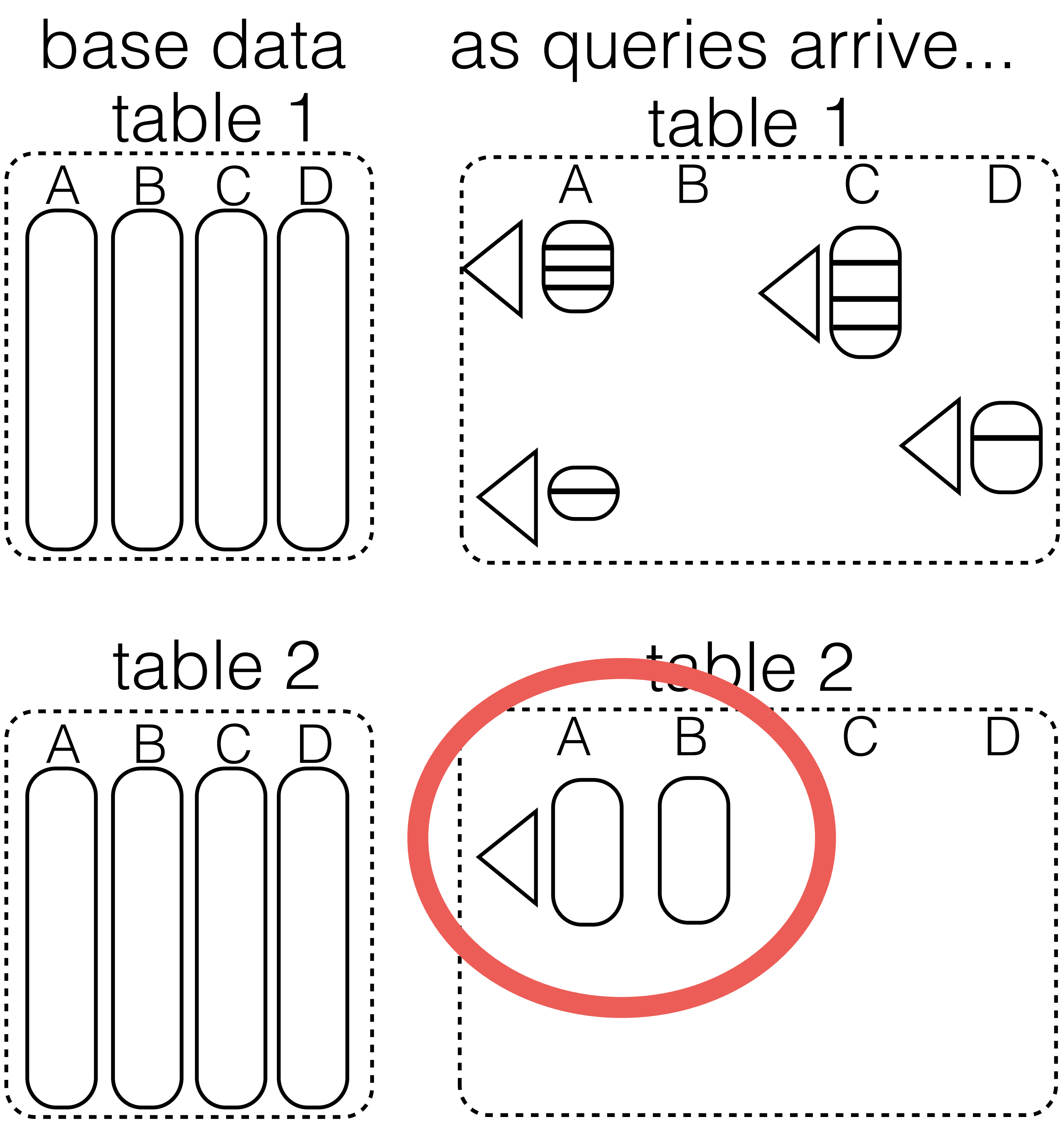
partial materialization

partial indexing

continuous adaptation

storage adaptation

cracking tangram



partial materialization

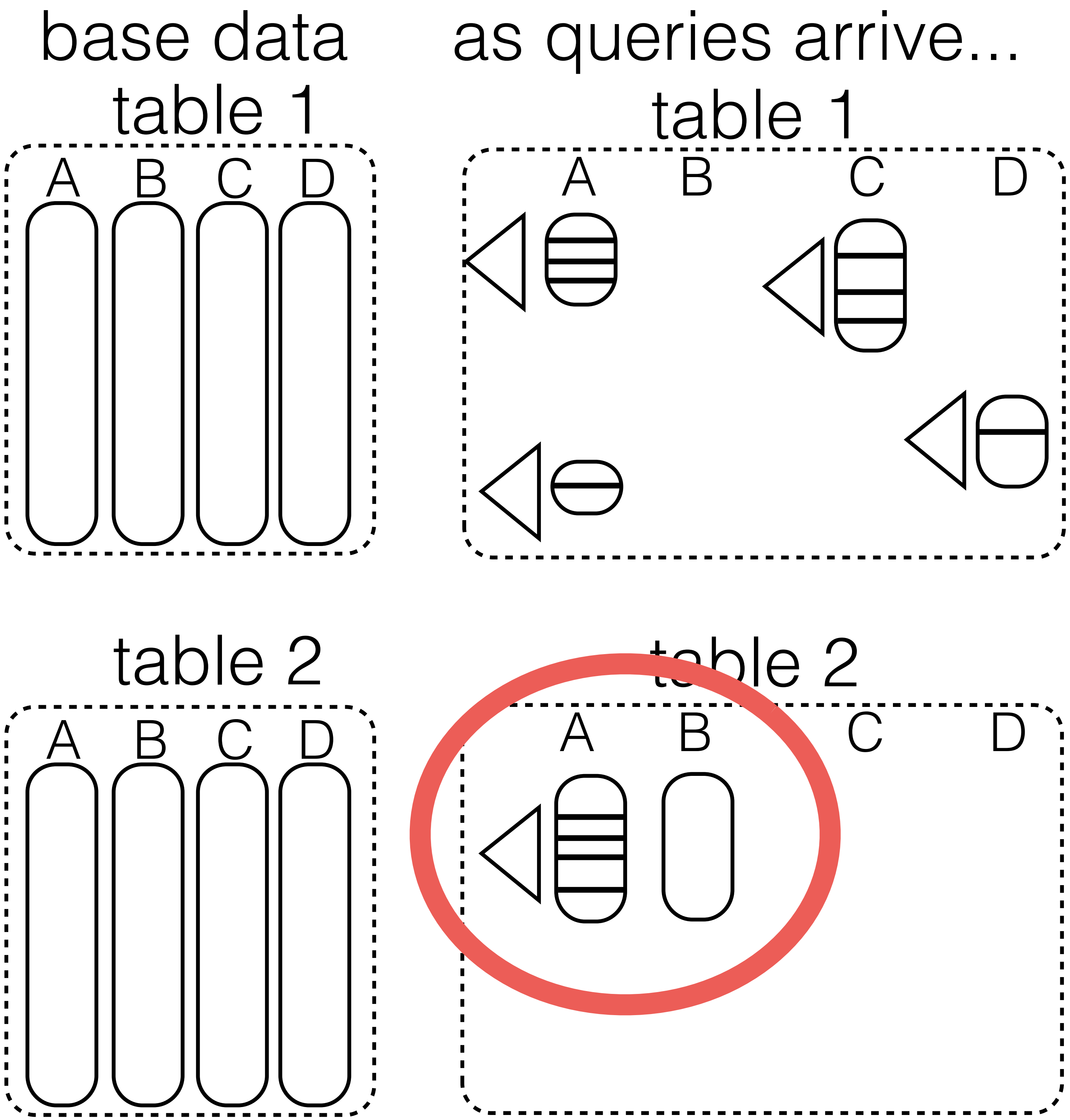
partial indexing

continuous adaptation

storage adaptation

no tuple reconstruction

cracking tangram



partial materialization

partial indexing

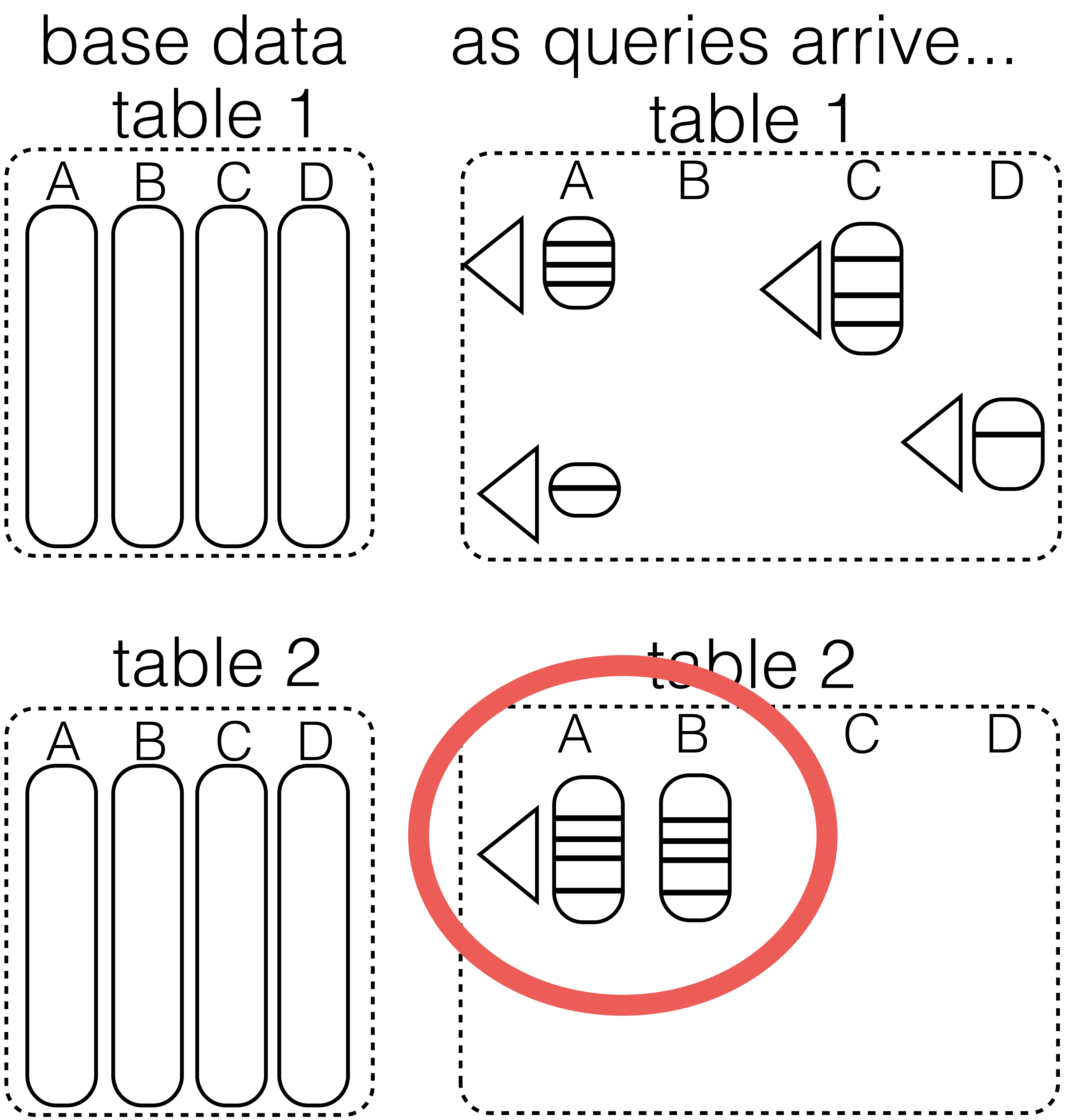
continuous adaptation

storage adaptation

no tuple reconstruction

adaptive alignment

cracking tangram



partial materialization

partial indexing

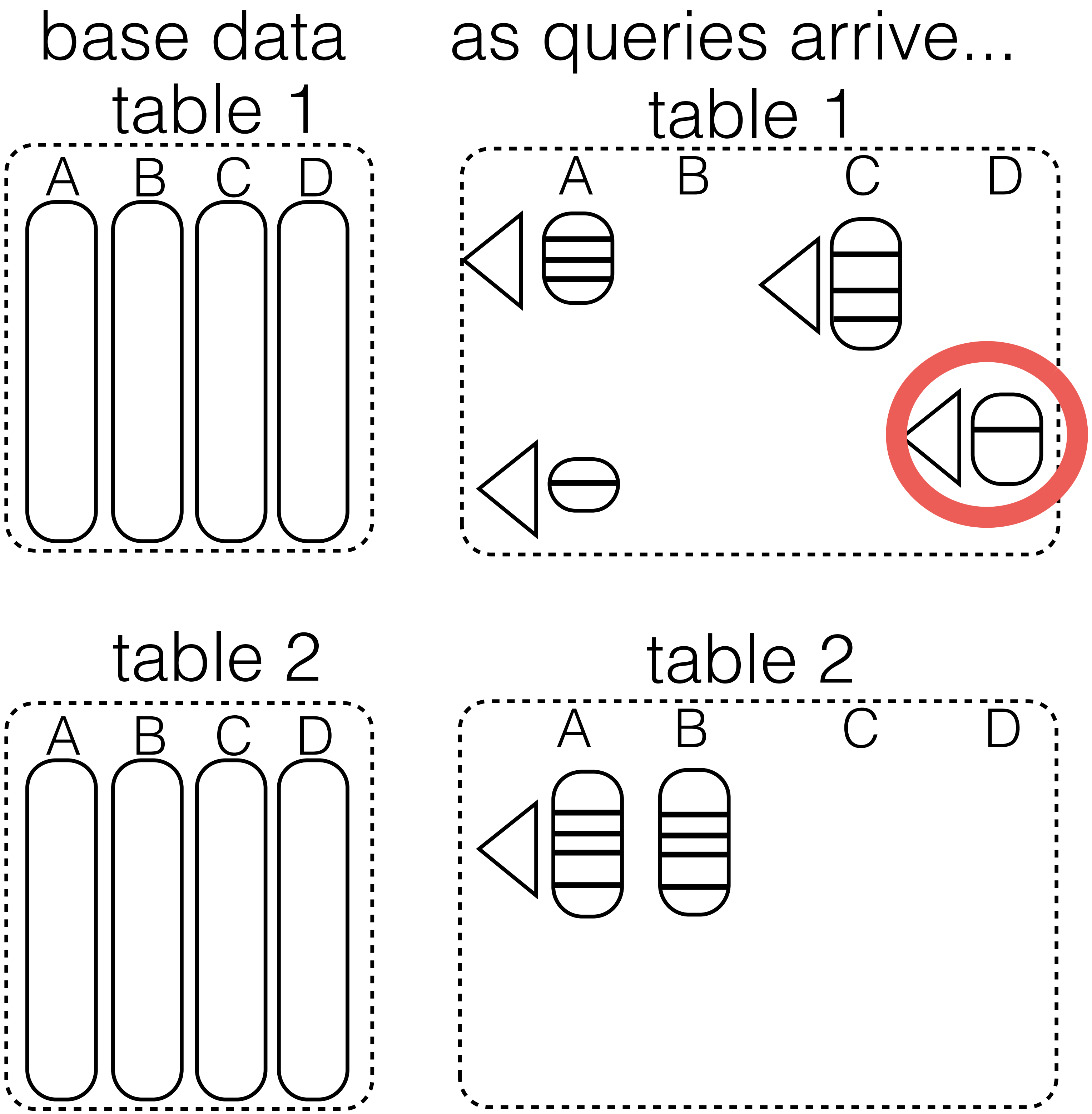
continuous adaptation

storage adaptation

no tuple reconstruction

adaptive alignment

cracking tangram



partial materialization

partial indexing

continuous adaptation

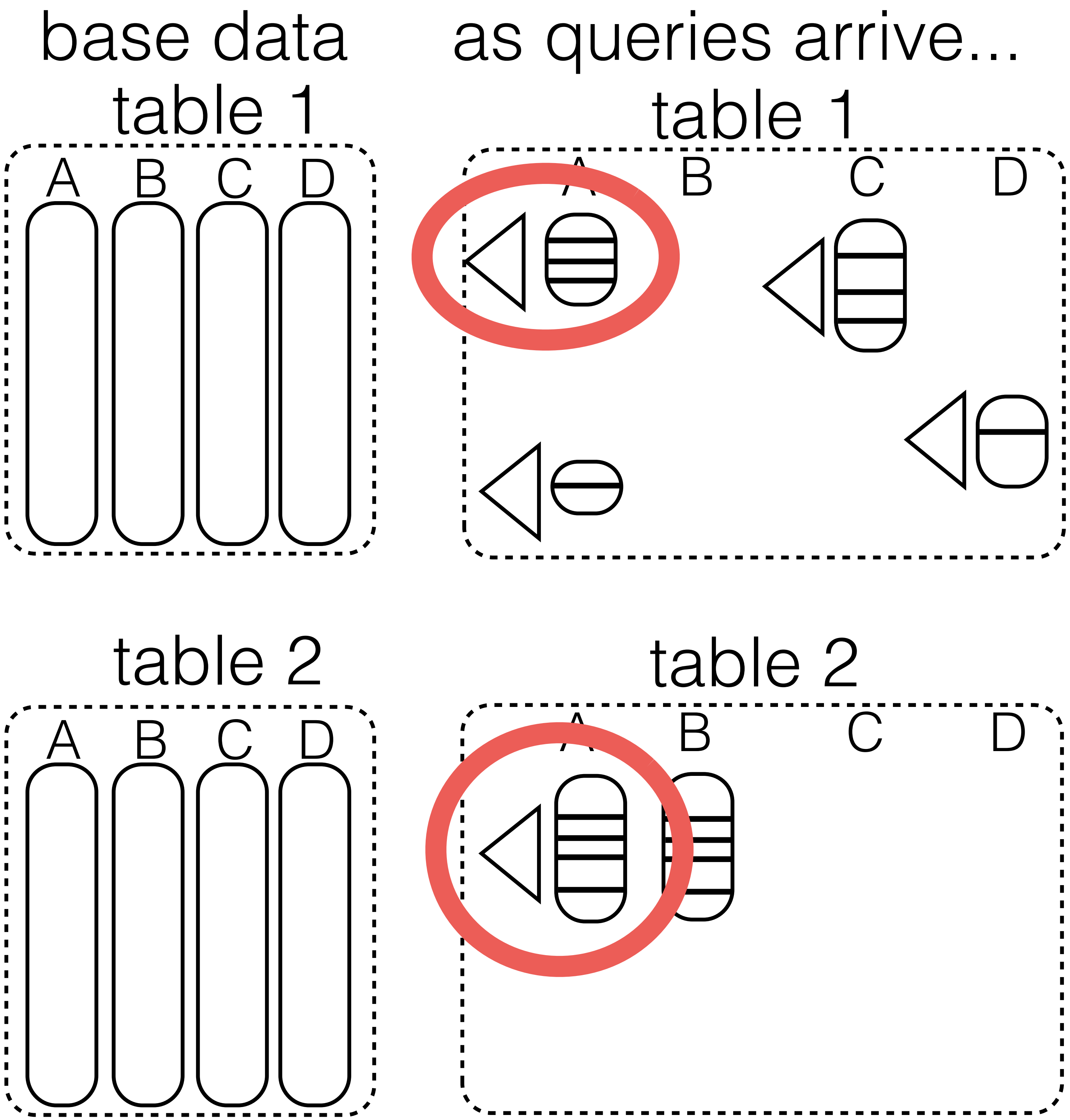
storage adaptation

no tuple reconstruction

adaptive alignment

sort in caches

cracking tangram



partial materialization

partial indexing

continuous adaptation

storage adaptation

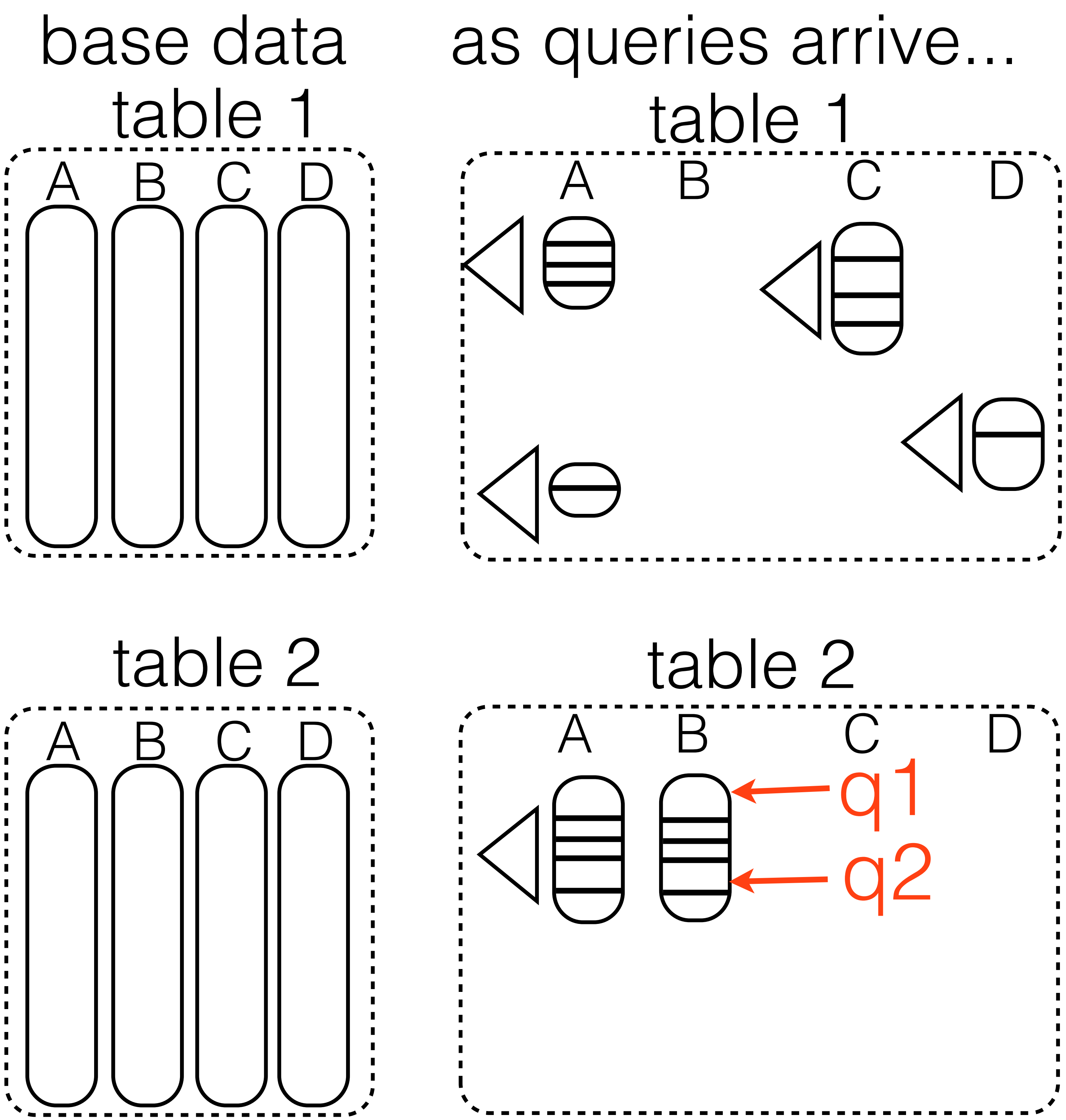
no tuple reconstruction

adaptive alignment

sort in caches

crack joins

cracking tangram



partial materialization

partial indexing

continuous adaptation

storage adaptation

no tuple reconstruction

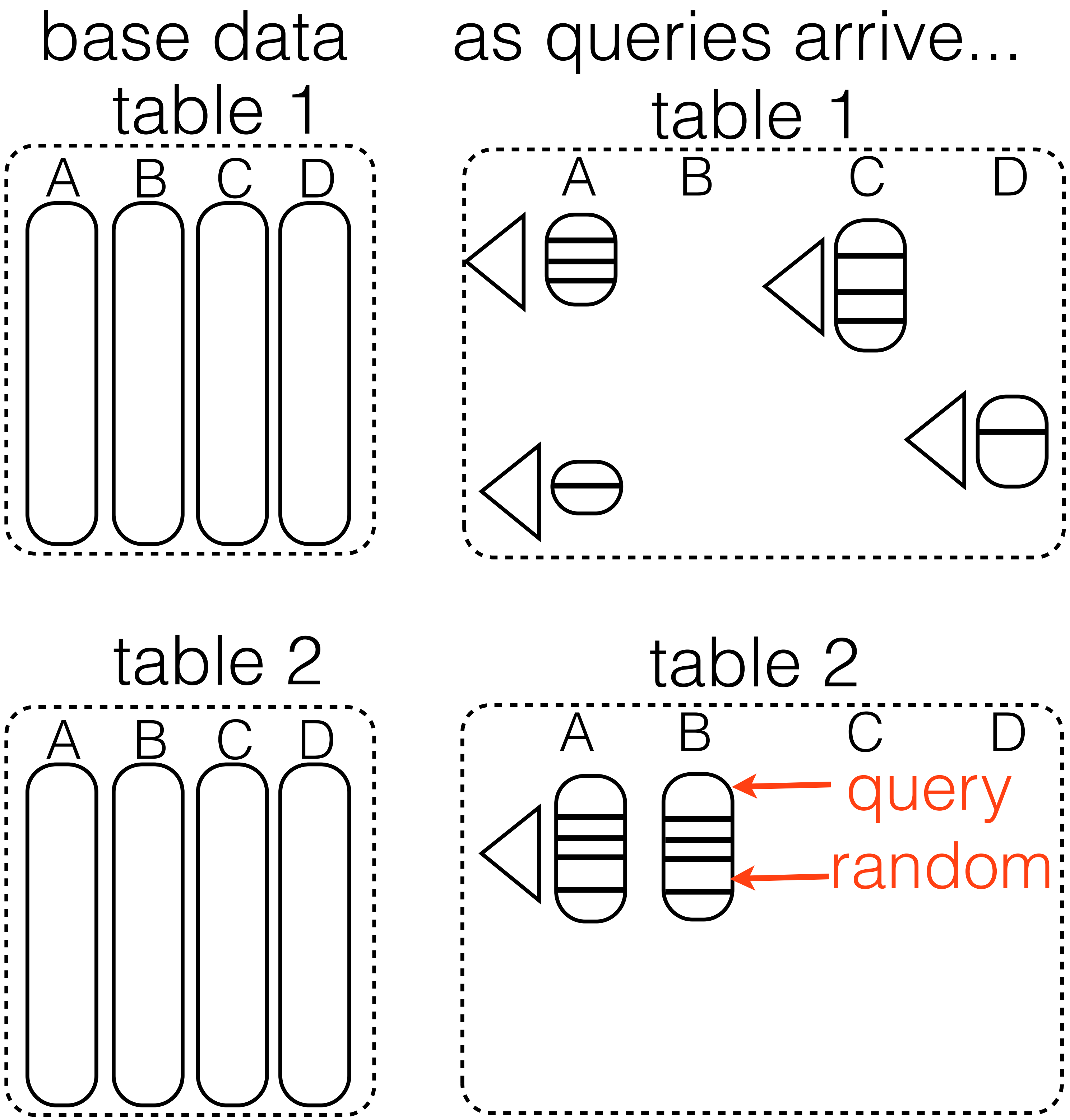
adaptive alignment

sort in caches

crack joins

lightweight locking

cracking tangram



partial materialization

partial indexing

continuous adaptation

storage adaptation

no tuple reconstruction

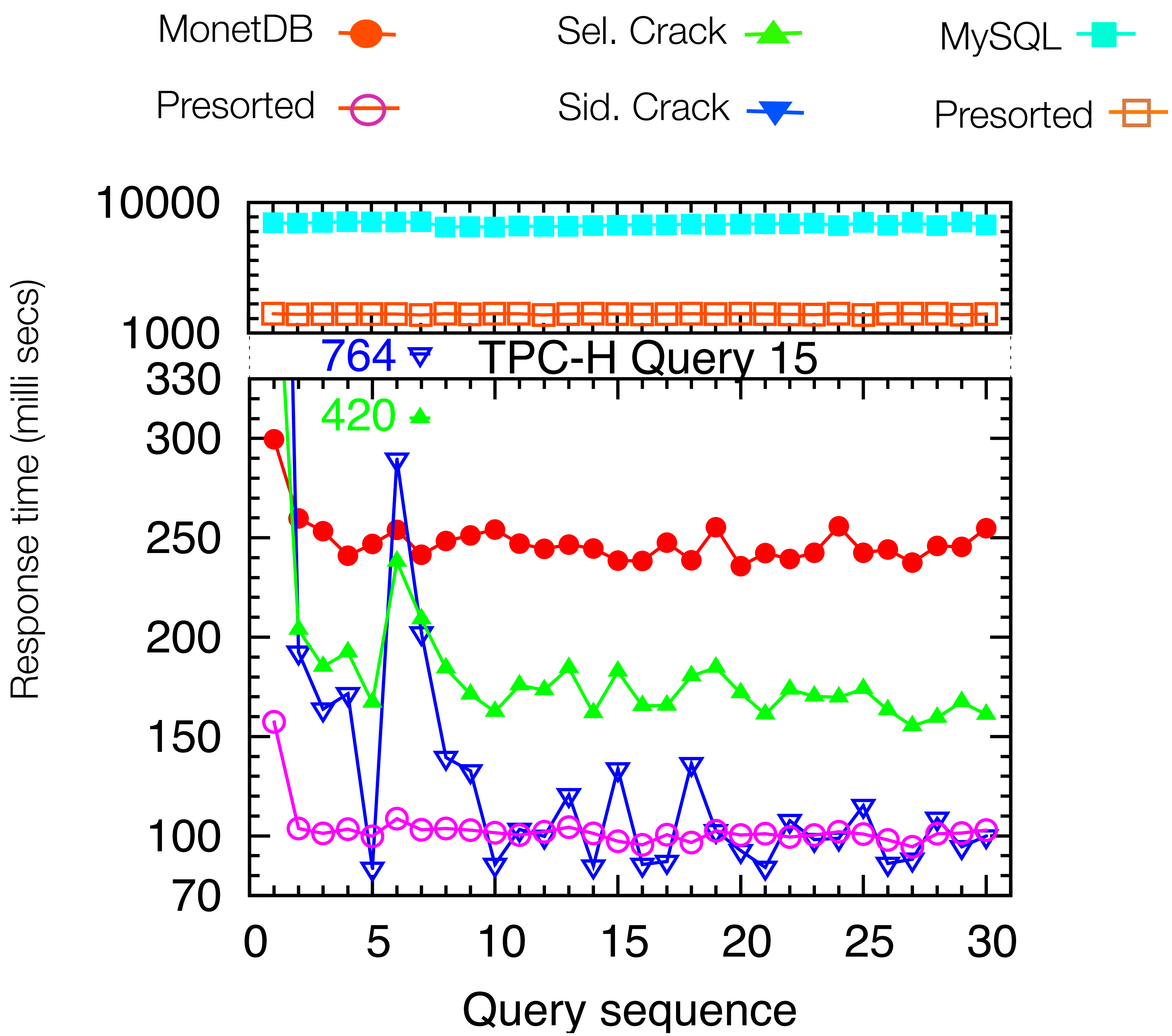
adaptive alignment

sort in caches

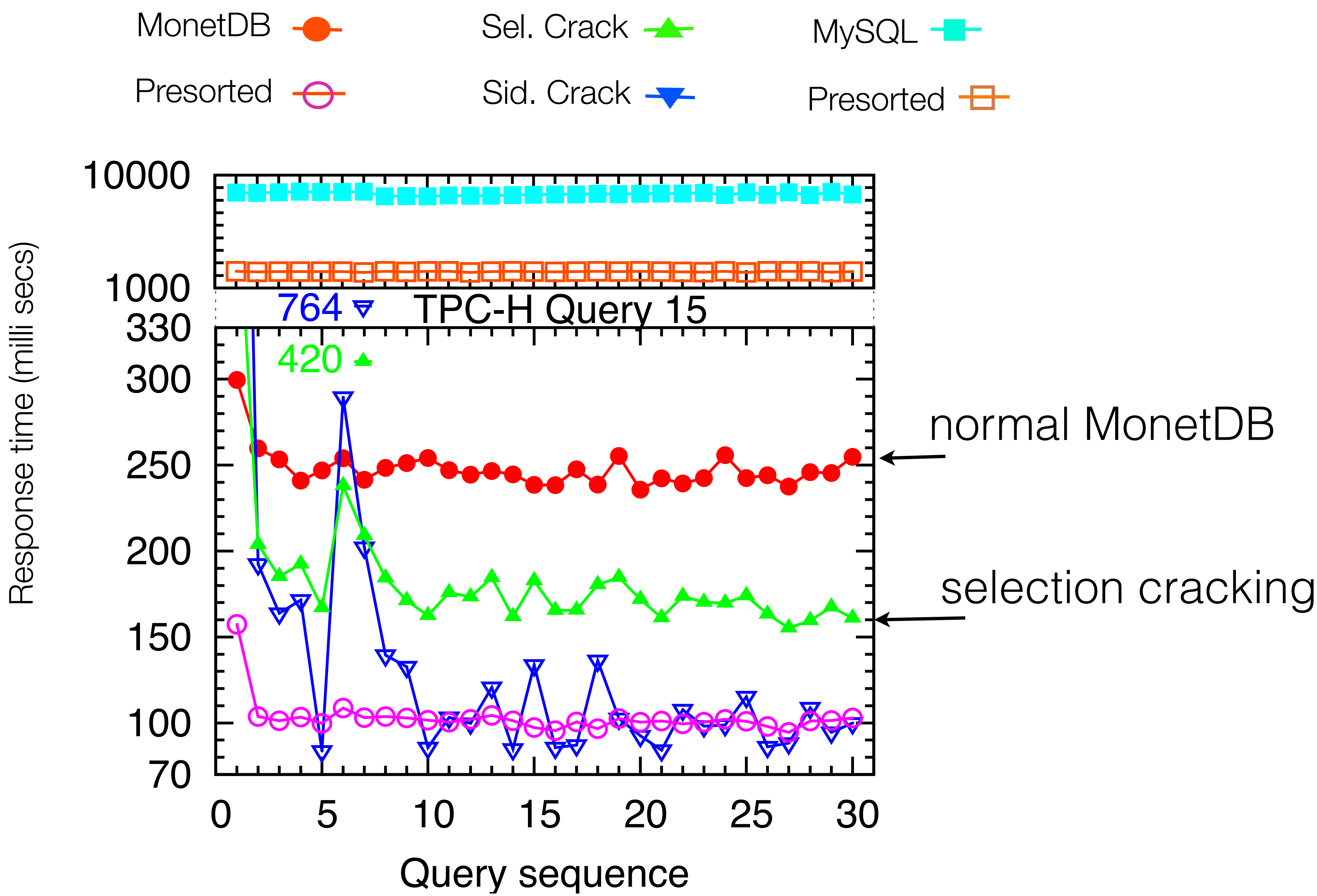
crack joins

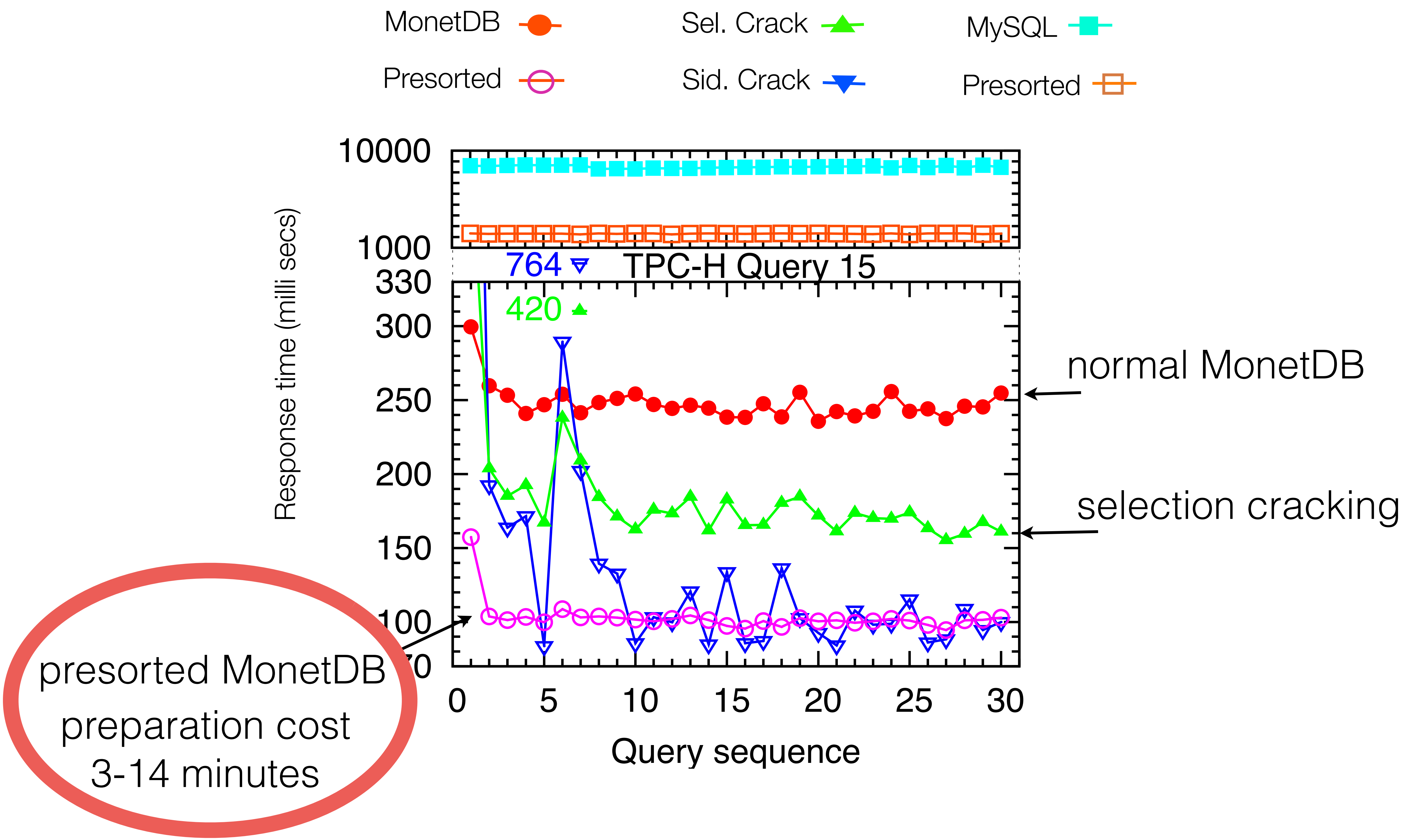
lightweight locking

stochastic cracking

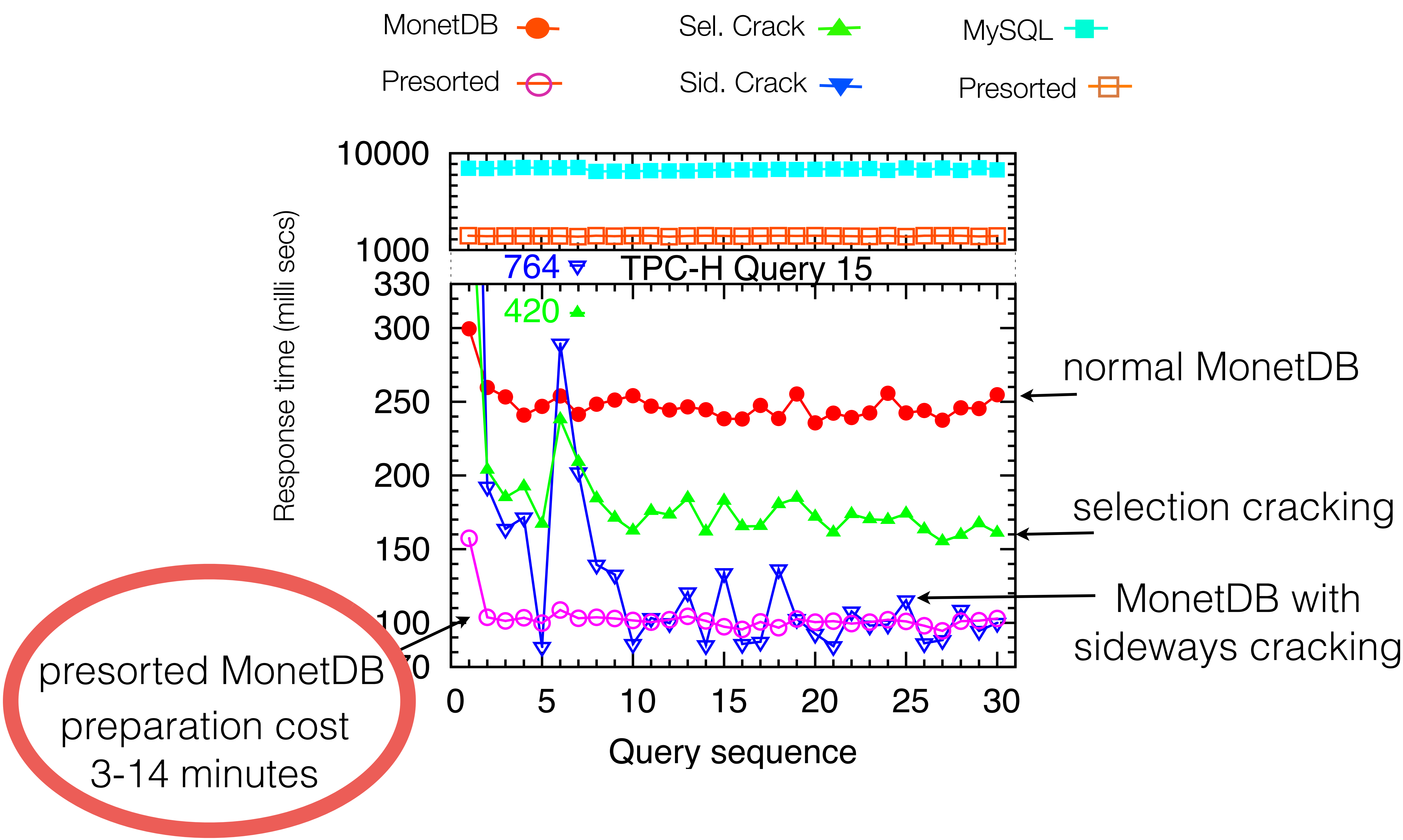


MonetDB
Sel. Crack
MySQL
Presorted
Sid. Crack
Presorted
Response time (milli secs)
10000
1000
330
300
250
200
150
100
70
764
TPC-H Query 15
420
0
5
10
15
20
25
30
Query sequence

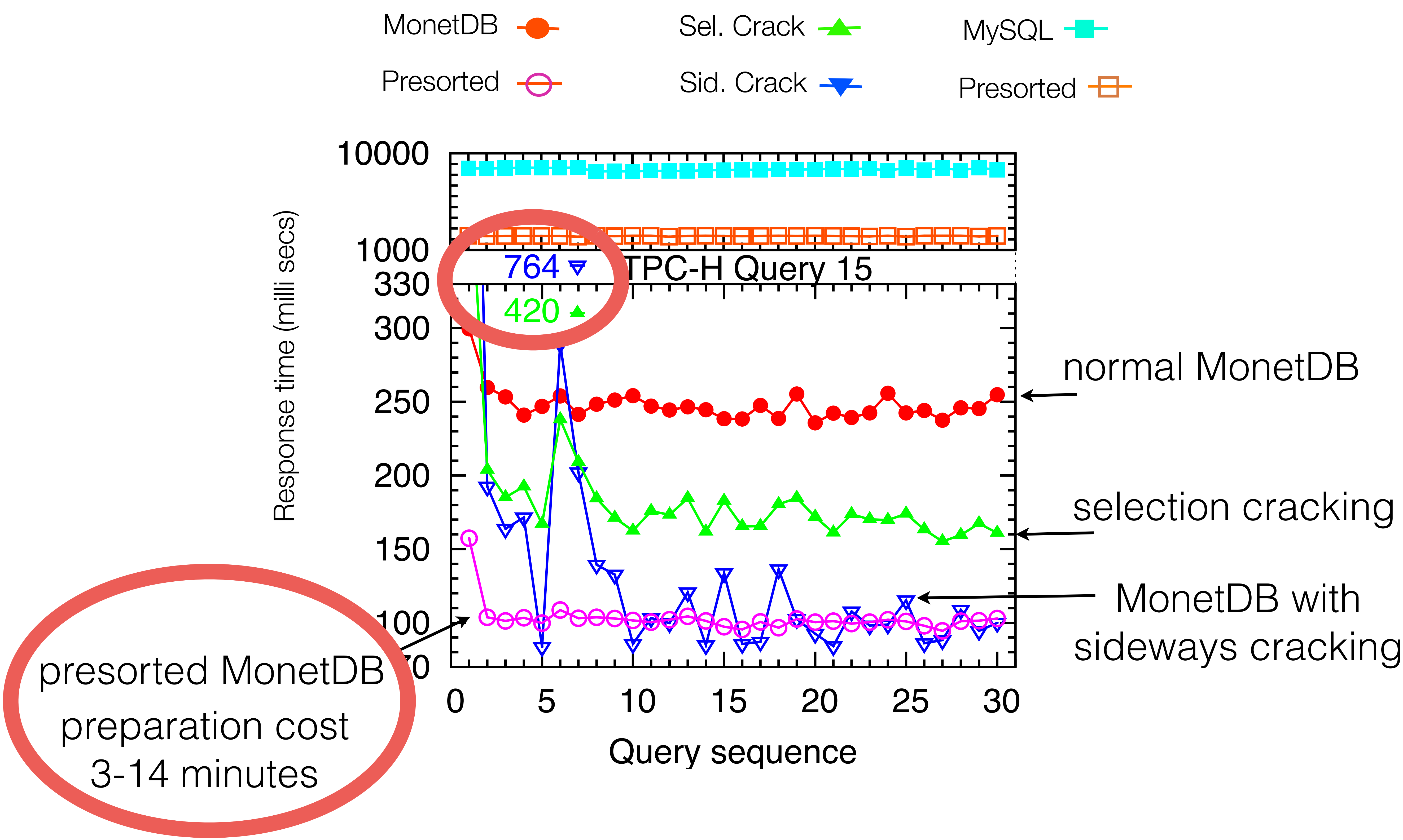

MonetDB
Sel. Crack
MySQL
Presorted
Sid. Crack
Presorted
10000
1000
330
300
250
200
150
100
70
764
TPC-H Query 15
420
0
5
10
15
20
25
30
Response time (milli secs)
Query sequence
normal MonetDB
selection cracking

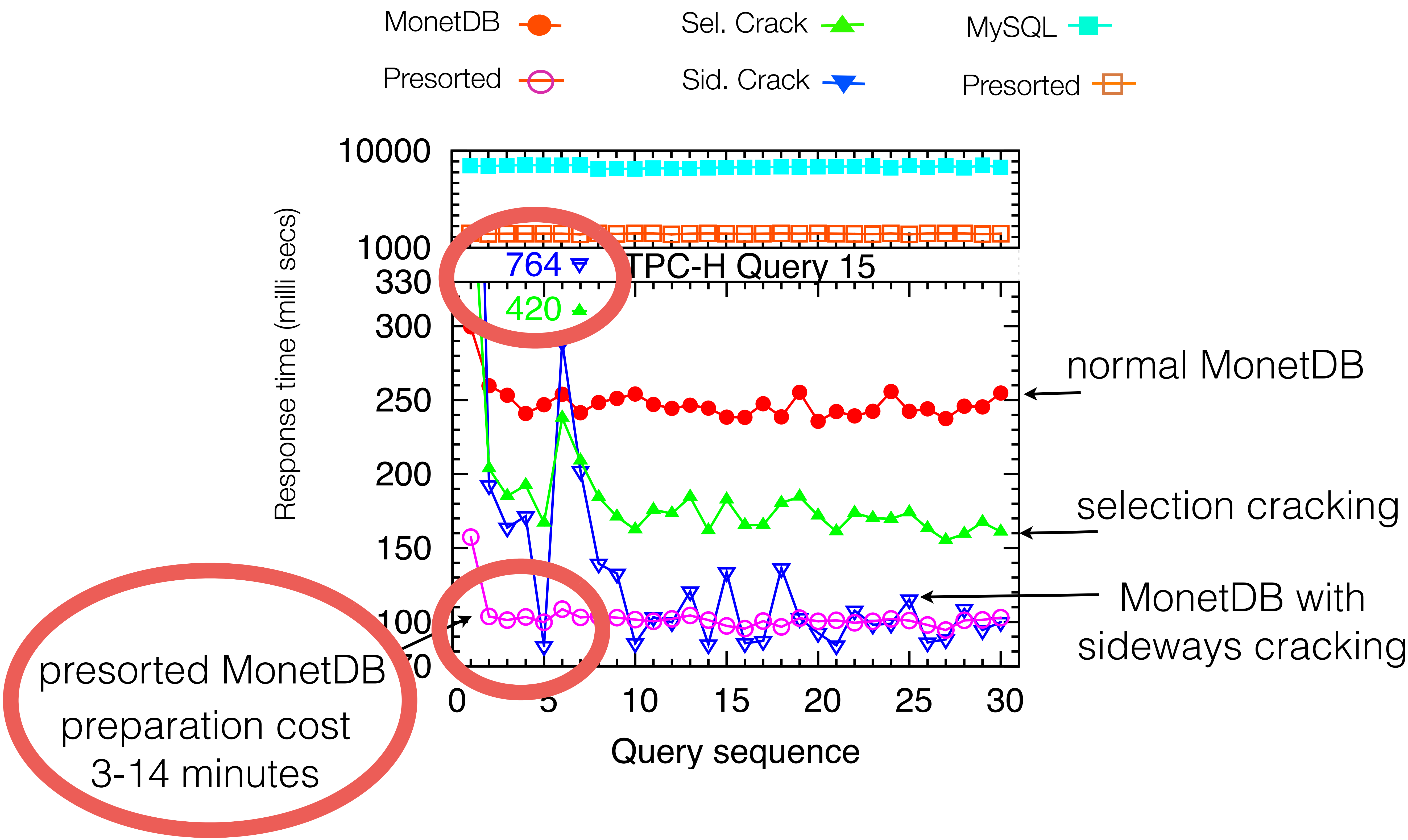


MonetDB
Sel. Crack
MySQL
Presorted
Sid. Crack
Presorted
10000
1000
330
300
250
200
150
100
764
TPC-H Query 15
420
Response time (milli secs)
0
5
10
15
20
25
30
Query sequence
normal MonetDB
selection cracking
presorted MonetDB
preparation cost
3-14 minutes

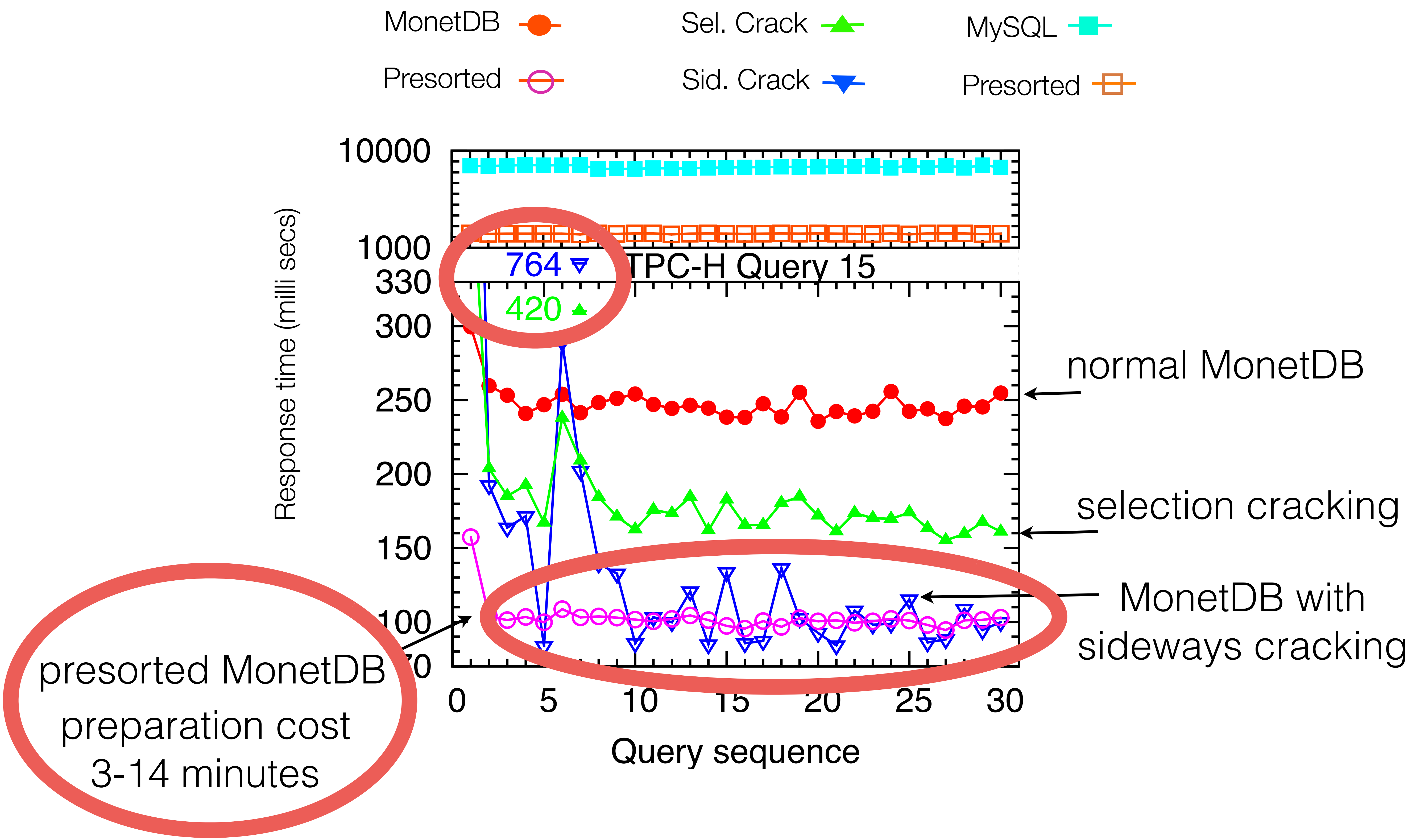


MonetDB
Sel. Crack
MySQL
Presorted
Sid. Crack
Presorted
10000
1000
330
300
250
200
150
100
764
TPC-H Query 15
420
0
5
10
15
20
25
30
Query sequence
Response time (milli secs)
normal MonetDB
selection cracking
MonetDB with
sideways cracking
presorted MonetDB
preparation cost
3-14 minutes


MonetDB
Presorted
Sel. Crack
Sid. Crack
MySQL
Presorted
10000
1000
330
300
250
200
150
100
764
420
TPC-H Query 15
Response time (milli secs)
0
5
10
15
20
25
30
Query sequence
normal MonetDB
selection cracking
MonetDB with
sideways cracking
presorted MonetDB
preparation cost
3-14 minutes



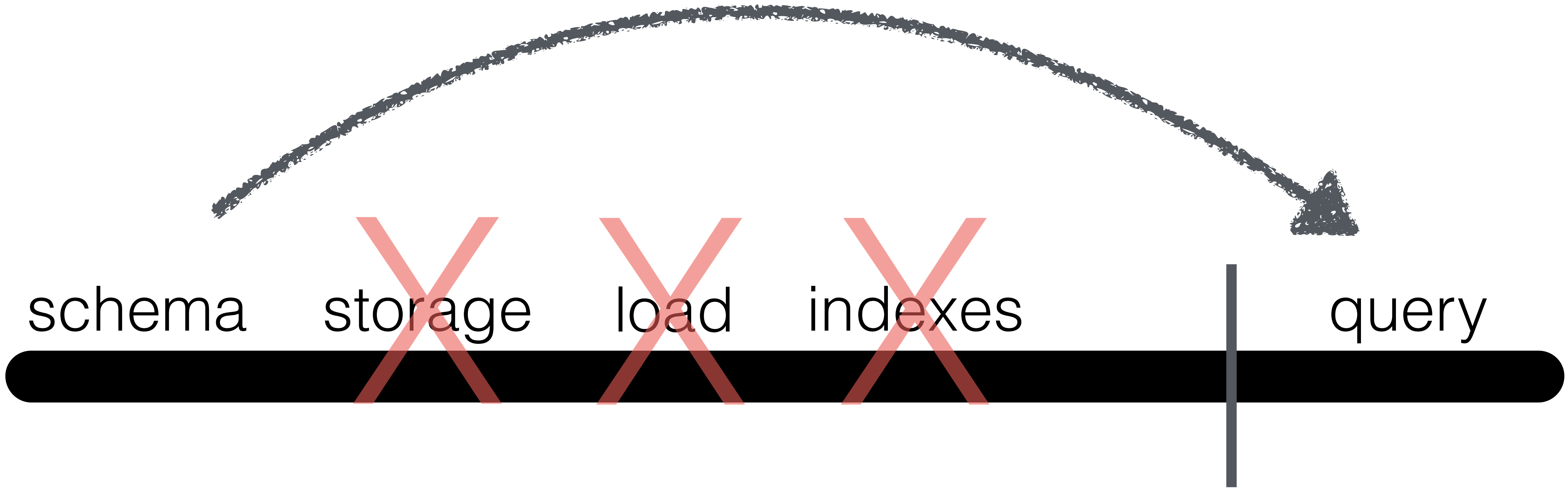
MonetDB
Sel. Crack
MySQL
Presorted
Sid. Crack
Presorted
10000
1000
330
300
250
200
150
100
70
764
420
TPC-H Query 15
Response time (milli secs)
0
5
10
15
20
25
30
Query sequence
normal MonetDB
selection cracking
MonetDB with
sideways cracking
presorted MonetDB
preparation cost
3-14 minutes

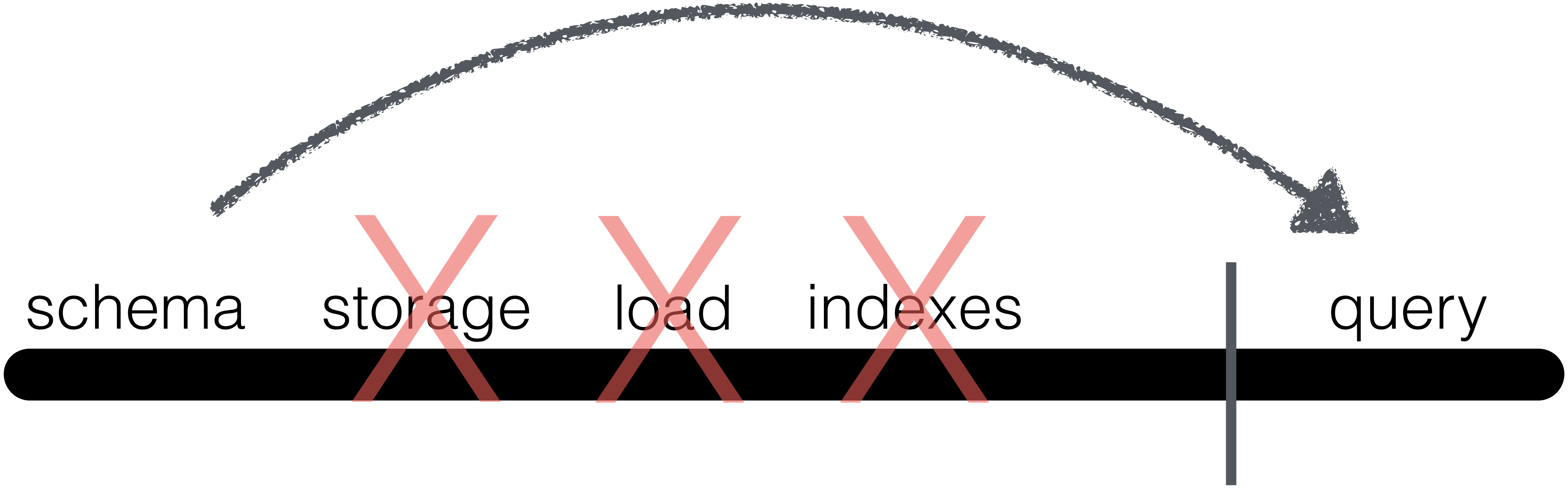

MonetDB
Presorted
Sel. Crack
Sid. Crack
MySQL
Presorted
10000
1000
330
300
250
200
150
100
70
764
420
TPC-H Query 15
Response time (milli secs)
0
5
10
15
20
25
30
Query sequence
normal MonetDB
selection cracking
MonetDB with
sideways cracking
presorted MonetDB
preparation cost
3-14 minutes

cracking on Skyserver (4TB)

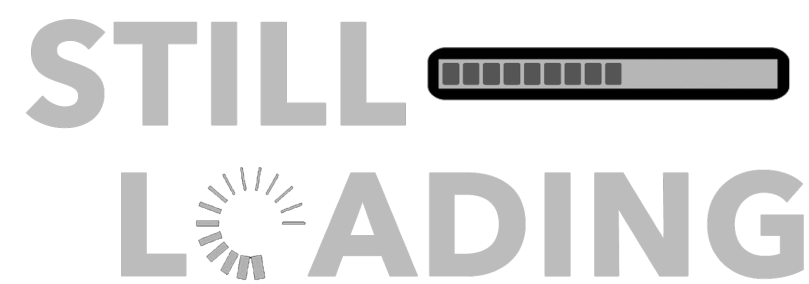
(Sloan Digital Sky Survey, www.sdss.org)

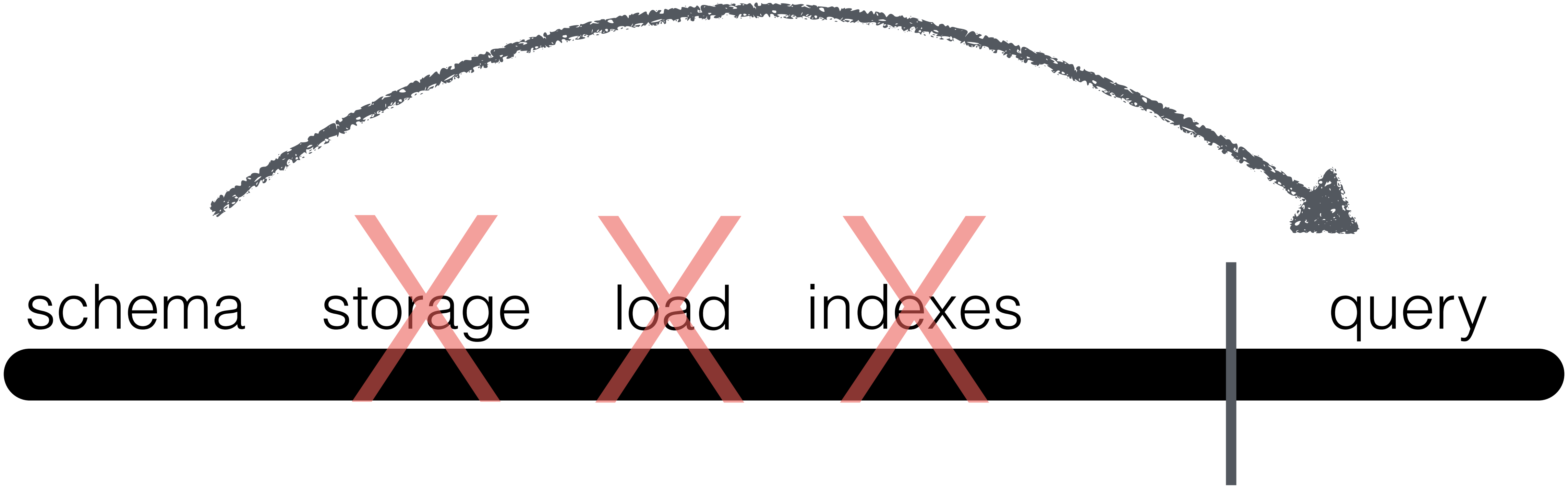
cracking answers 160.000 queries
while full indexing is still half way creating one index

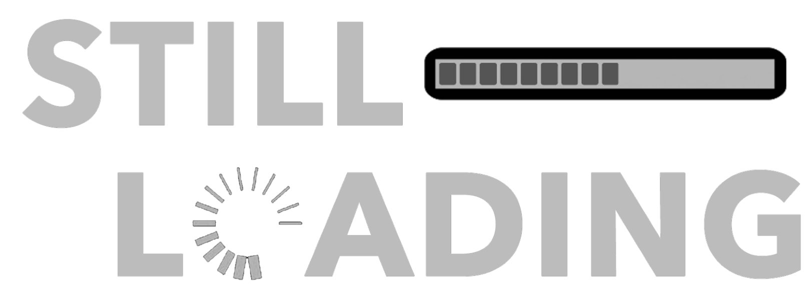

schema
storage
load
indexes
query



adaptive loading (NoDB, CIDR11/SIGMOD12)

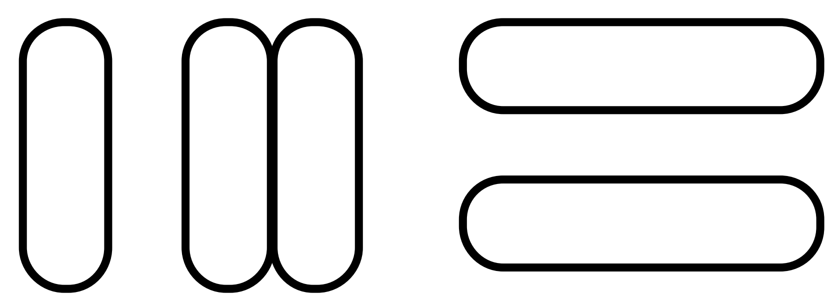


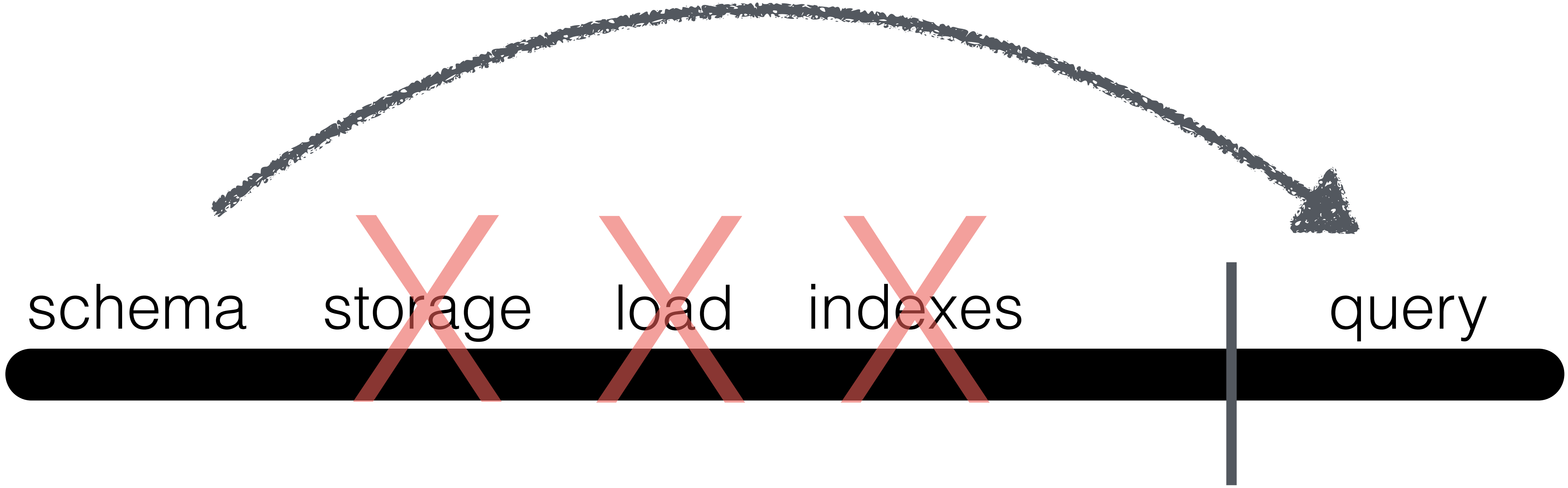




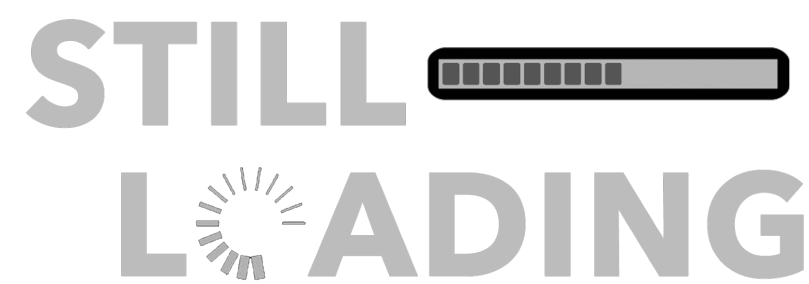
adaptive loading (NoDB, CIDR11/SIGMOD12)

adaptive storage (H20, SIGMOD14)

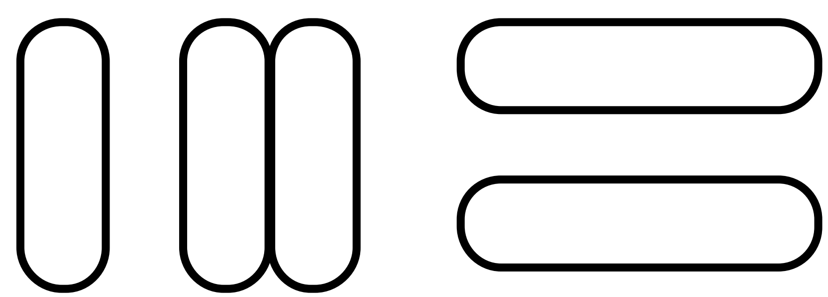




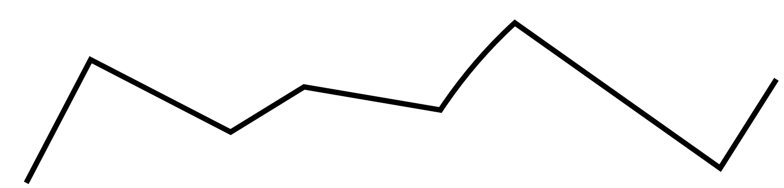
adaptive loading (NoDB, CIDR11/SIGMOD12)

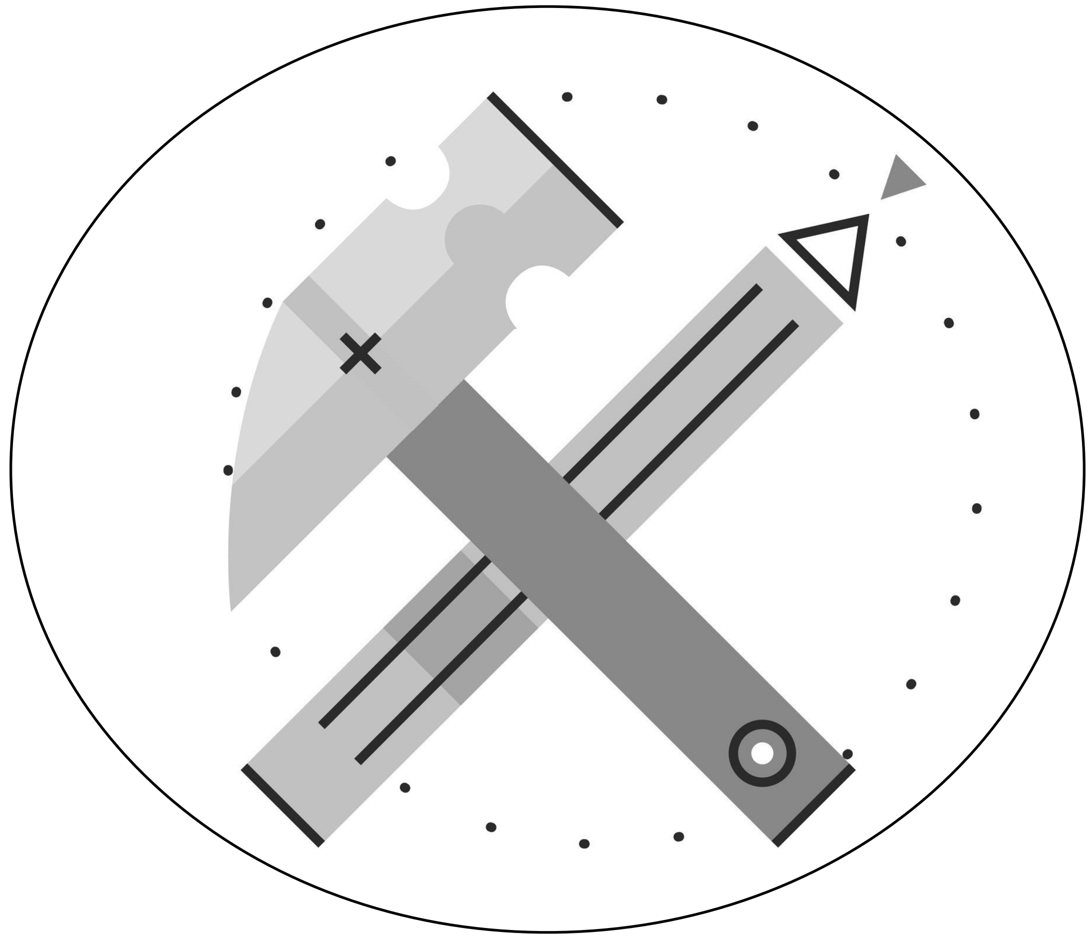


adaptive storage (H20, SIGMOD14)



adaptive time series indexing (ADS, SIGMOD14)





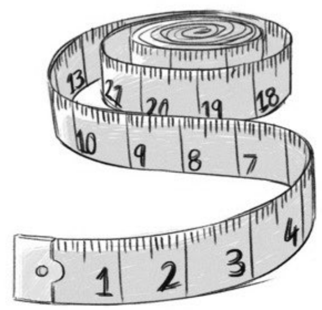
data systems that are easy to design

(storage, data flow, algorithms, tuning, etc)

HARVARD School of Engineering and Applied Sciences Stratos Idreos

One size does **not** fit all

+



Custom solutions are needed for optimal performance

+

Solutions need to be tuned

=

Bootstrapping new systems is expensive and time-consuming

e.g., column-stores:
first ideas in 80s,
first advanced architectures in 90s,
first rather complete designs in early 2000s,
industry adoption 2010+
still no indexing, cost based optimizations, …

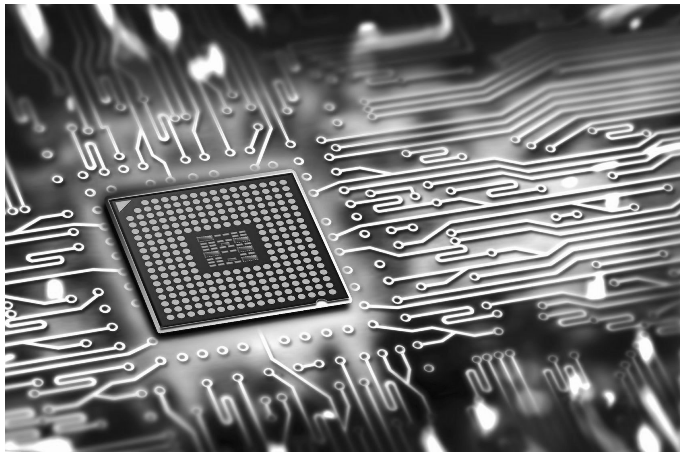
conflicting goals moving target

(hardware and requirements change continuously and rapidly)

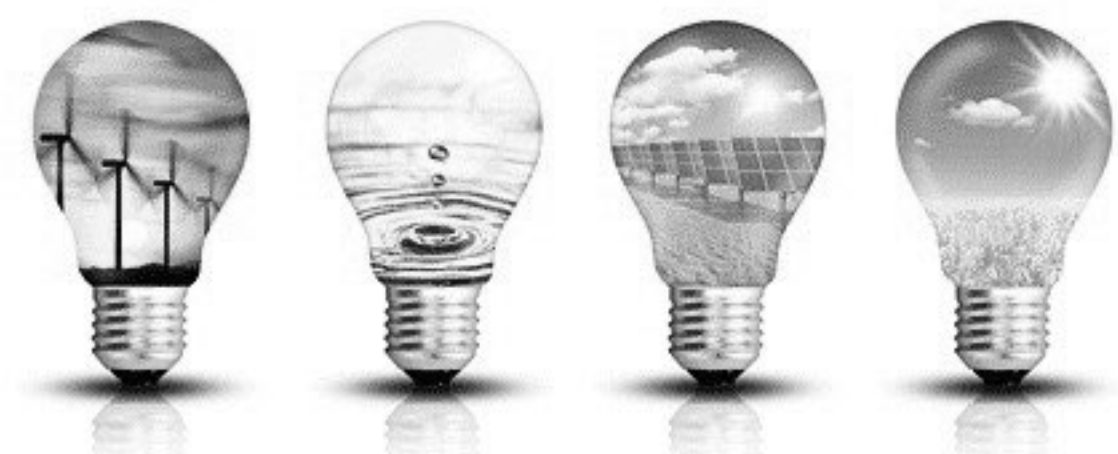
application requirements

performance

budget



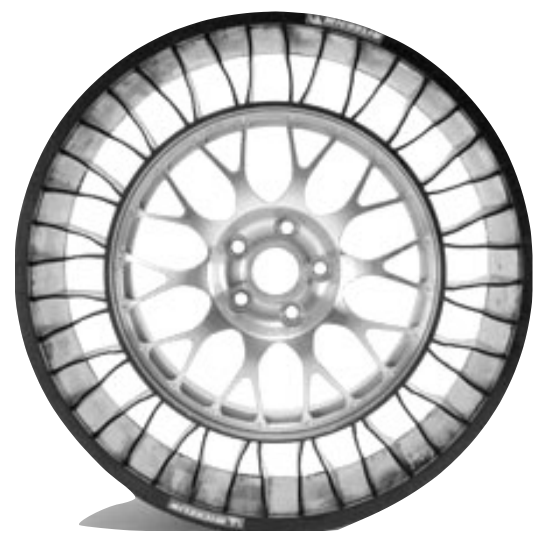
hardware



energy profile

data systems design (and research) is kind of an **art**

disk



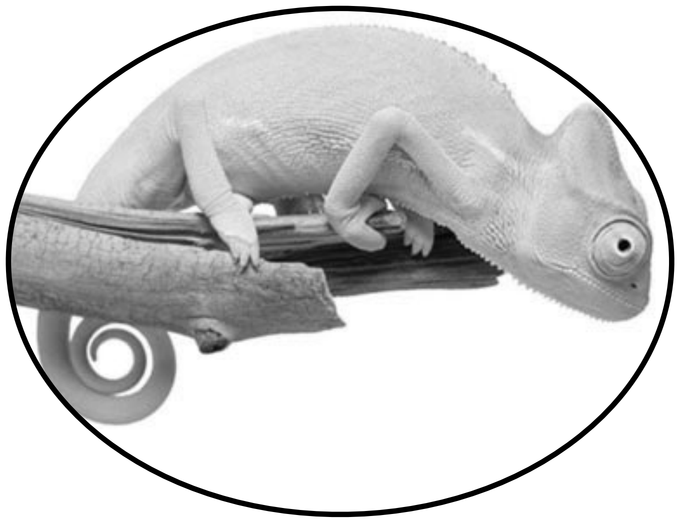
memory

flash

. . .

self-designing data systems

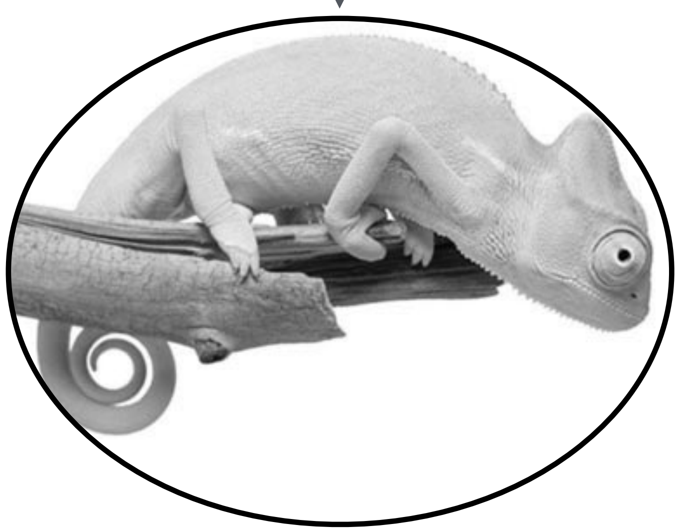
data+queries+hardware



data system

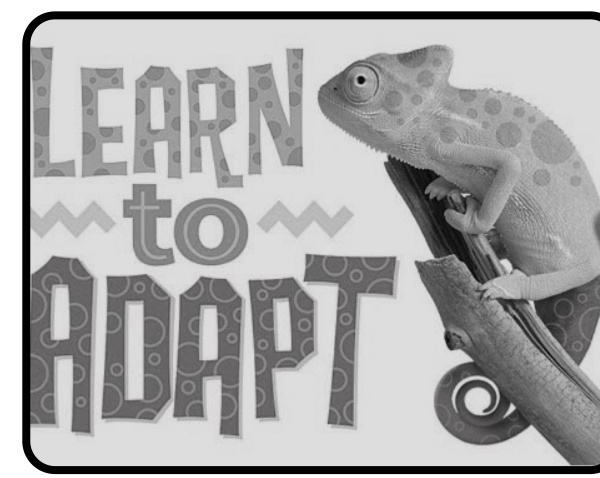
self-designing data systems

data+queries+hardware



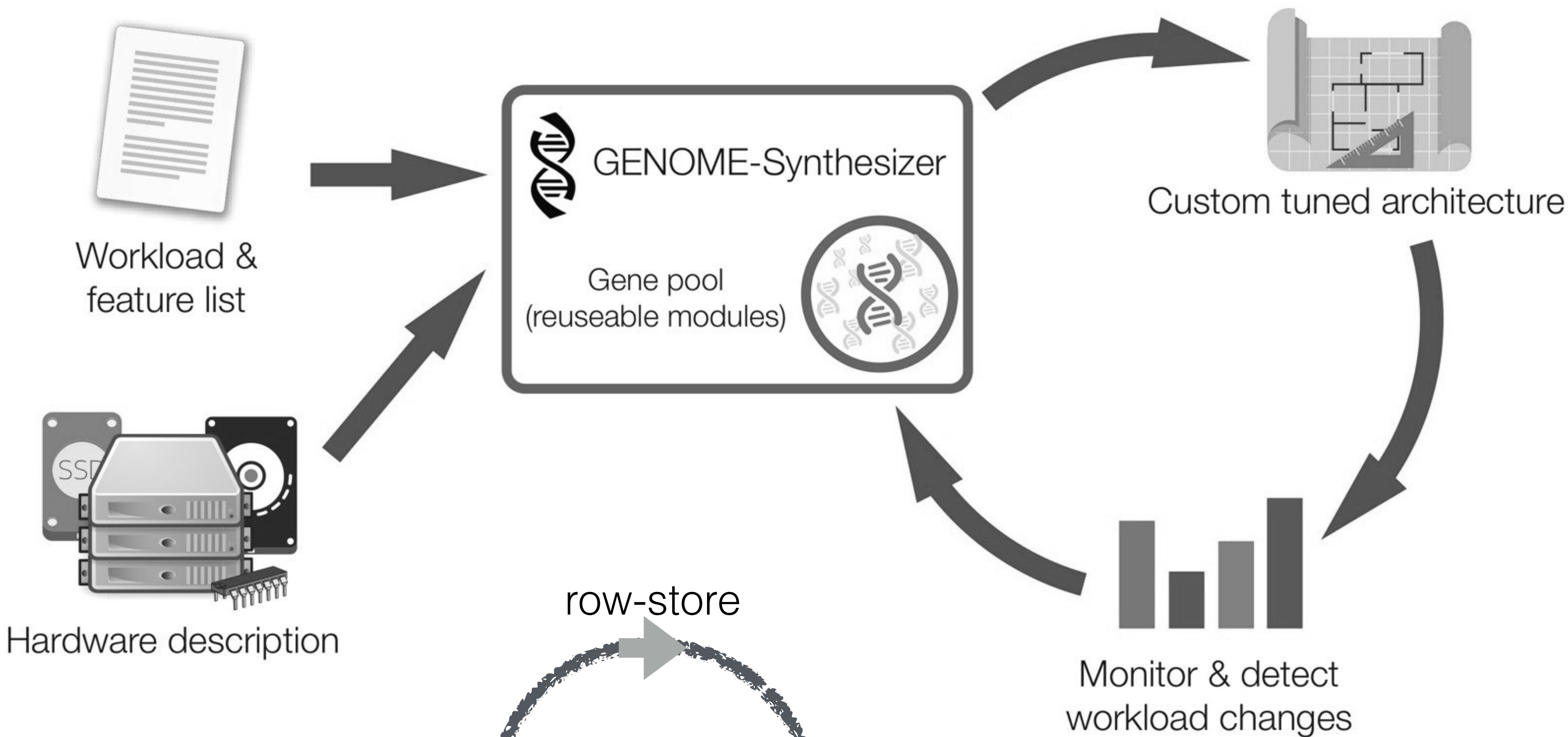
data system

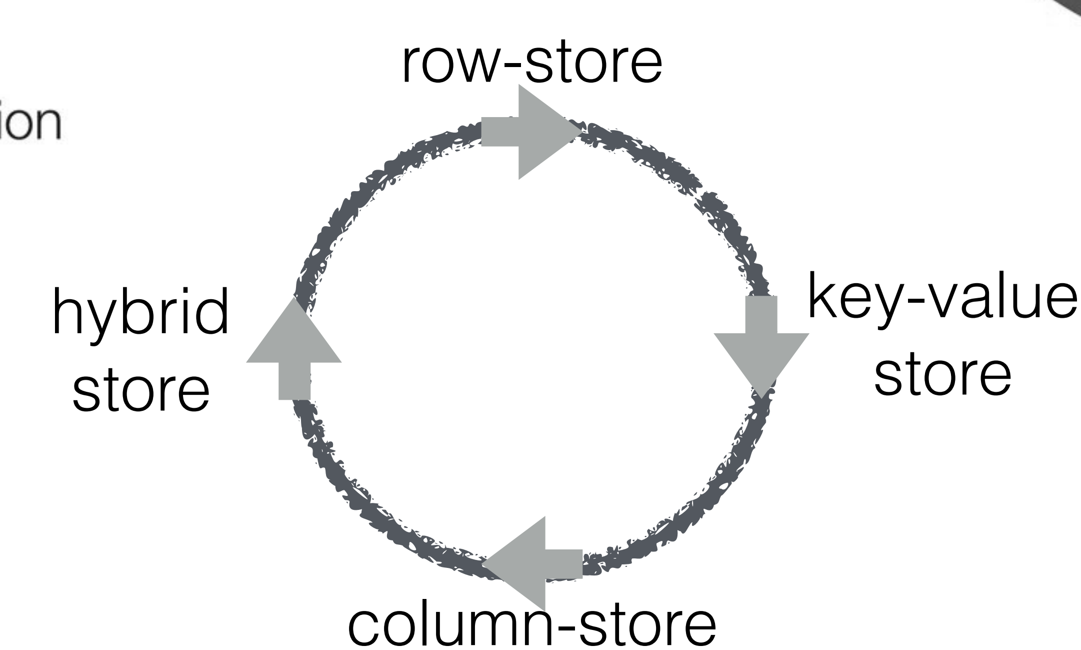
easy to design



adapt to environment

adaptivity across architecture borders



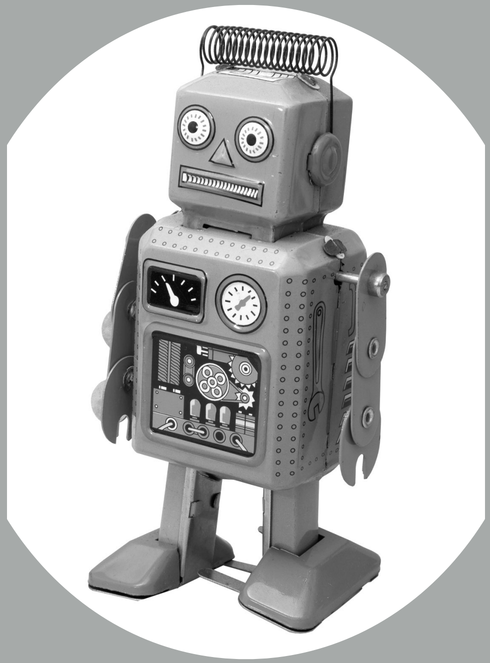


HARVARD
School of Engineering
and Applied Sciences
Stratos Idreos

data systems that
are easy to use

dbTouch

show me something interesting



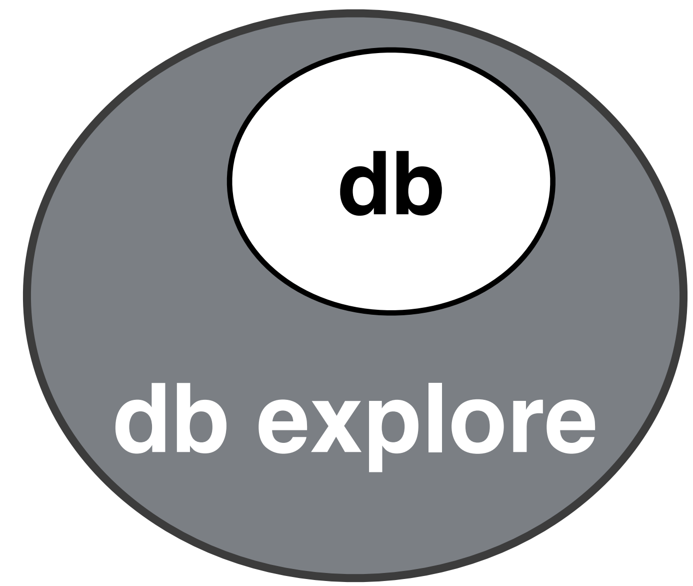
— DATA

Queriosity

HARVARD
School of Engineering
and Applied Sciences
Stratos Idreos

data systems today

allow us to answer queries fast



data systems tomorrow

should allow us to find fast which queries to ask instead of making fixed decisions

every query is treated as an advice on how data should be stored

http://daslab.seas.harvard.edu/

Martin Kersten
Stefan Manegold
Goetz Graefe
Harumi Kuno
Anastasia Ailamaki
Themis Palpanas
Eleni Petraki

Ioannis Alagiannis
Miguel Branco
Renata Borovica
Erietta Liarou
Felix Halim
Ronald Yap
Panos Karras

Kostas Zoumpatianos
Manos Athanassoulis
Lukas Maas
Abdul Wasay
Mike Kester
Dhruv Gupta

HARVARD
School of Engineering
and Applied Sciences

Stratos Idreos

http://daslab.seas.harvard.edu/

thank you!

Martin Kersten
Stefan Manegold
Goetz Graefe
Harumi Kuno
Anastasia Ailamaki
Themis Palpanas
Eleni Petraki

Ioannis Alagiannis
Miguel Branco
Renata Borovica
Erietta Liarou
Felix Halim
Ronald Yap
Panos Karras

Kostas Zoumpatianos
Manos Athanassoulis
Lukas Maas
Abdul Wasay
Mike Kester
Dhruv Gupta

HARVARD
School of Engineering
and Applied Sciences

Stratos Idreos