

# CS 561: Data Systems Architectures

Prof. Manos Athanassoulis

[mathan@bu.edu](mailto:mathan@bu.edu)

<https://bu-disc.github.io/CS561/>

no   
smartphones

no   
laptop

**Why?**

there is enough evidence that laptops and phones slow you down

Today

big data

data-driven world

data systems

*which are the driving trends?*

*why do we need new designs?*

**CS 561 goals & logistics**



I want you to speak up!  
[and you can always interrupt me]

# CS 561 philosophy

cutting-edge research

question everything (to understand it better!)

*There are no stupid questions!*

interactive & collaborative

projects, presentations, labs, OH



# Understanding a design/system/algorithm ...

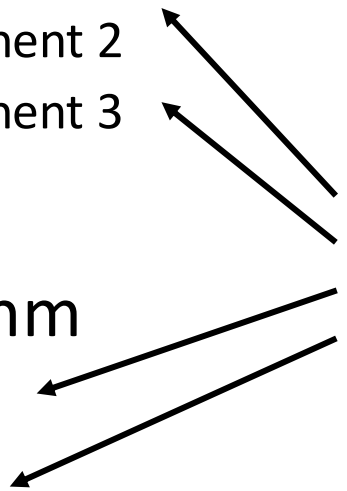
## system

- component 1
- component 2
- component 3

## algorithm

- step 1
- step 2
- step 3

why?  
why not?



understanding all steps and all decisions  
helps us see the ***big picture***  
and do **good research!**

(otherwise, we make ad hoc choices!)



# Ask Questions!

... and answer my questions!

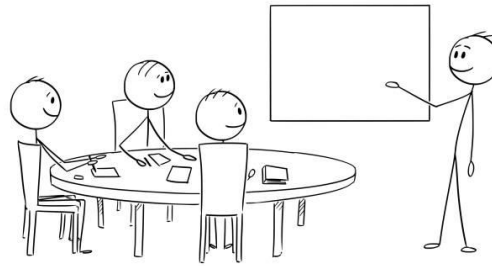
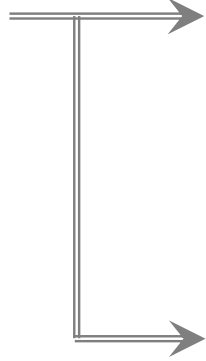
our **main goal** is to have **interesting discussions** that will help to gradually understand what the material discusses

**(it's ok if not everything is clear, as long as you have questions!)**

# What do we do in this class?



reading papers



presentations



technical questions



projects

# Reading Papers



every class **1-2 papers to discuss** in detail

*in some classes the discussion will be led by a group of students*

*so that, each student will present one paper during the semester*

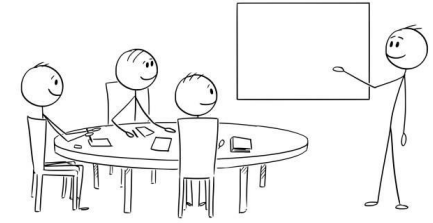
(background papers also available to provide more details)

read all of them!

answer technical questions as 10-min quizzes (for a subset of the papers)



# Paper Presentation / Student Discussion



**3-5 courses (later in the semester) will be entirely led by students!**

**A group of ~10 students will be split in three subgroups:**

- **presenters**, who will develop the slides and deliver the presentation
- **critics**, who will raise appropriate doubts or question about the paper
- **proponents**, who will explain how this paper addresses these

Note that the three subgroups can coordinate and prepare together.

Each student must participate once in a presentation/discussion in (any) one of the three roles

Prepare slides at least a **week before your presentation + discuss them in OH**



# Reading a Research Paper

learn

## **review**

what is the problem & why it is important?  
why is it hard & why older approaches are not enough?  
what is the key idea and why it works?

critic

what is missing and how can we improve this idea?  
does the paper support its claims?  
possible next steps of the work presented in the paper?

## **additional technical questions**

to make sure the heart of the paper is clearly understood

remember, this will help us do **good research!**

# Projects

## project 0

A small implementation project  
to sharpen dev skills

independent project



Due on Jan 31, 2025

***AND***

## project 1

A medium project to give you a flavor of  
large-scale production system

groups of 3



Due on Feb 14, 2025

# Projects

***AND***

## **project 0**

A small implementation project  
to sharpen dev skills

independent project



Due on Jan 31, 2025

## **project 1**

A medium project to give you a flavor of  
large-scale production system

groups of 3



Due on Feb 14, 2025

# Projects

# systems project

groups of 3

# implementation-heavy C/C++ project

[illegible]

**OR**

# research project

groups of 3

pick a subject (list will be available)

## design & analysis

## experimentation



# Projects

# systems project

groups of 3

# implementation-heavy C/C++ project

```

000 cout<<endl<<endl<<"Iterations: " << X11<<"X12: " << X10;
001 //
002 //
003 //
004 //
005 //
006 //
007 //
008 //
009 //
010 //
011 //
012 //
013 //
014 //
015 //
016 //
017 //
018 //
019 //
020 //
021 //
022 //
023 //
024 //
025 //
026 //
027 //
028 //
029 //
030 //
031 //
032 //
033 //
034 //
035 //
036 //
037 //
038 //
039 //
040 //
041 //
042 //
043 //
044 //
045 //
046 //
047 //
048 //
049 //
050 //
051 //
052 //
053 //
054 //
055 //
056 //
057 //
058 //
059 //
060 //
061 //
062 //
063 //
064 //
065 //
066 //
067 //
068 //
069 //
070 //
071 //
072 //
073 //
074 //
075 //
076 //
077 //
078 //
079 //
080 //
081 //
082 //
083 //
084 //
085 //
086 //
087 //
088 //
089 //
090 //
091 //
092 //
093 //
094 //
095 //
096 //
097 //
098 //
099 //
100 //
101 //
102 //
103 //
104 //
105 //
106 //
107 //
108 //
109 //
110 //
111 //
112 //
113 //
114 //
115 //
116 //
117 //
118 //
119 //
120 //
121 //
122 //
123 //
124 //
125 //
126 //
127 //
128 //
129 //
130 //
131 //
132 //
133 //
134 //
135 //
136 //
137 //
138 //
139 //
140 //
141 //
142 //
143 //
144 //
145 //
146 //
147 //
148 //
149 //
150 //
151 //
152 //
153 //
154 //
155 //
156 //
157 //
158 //
159 //
160 //
161 //
162 //
163 //
164 //
165 //
166 //
167 //
168 //
169 //
170 //
171 //
172 //
173 //
174 //
175 //
176 //
177 //
178 //
179 //
180 //
181 //
182 //
183 //
184 //
185 //
186 //
187 //
188 //
189 //
190 //
191 //
192 //
193 //
194 //
195 //
196 //
197 //
198 //
199 //
200 //
201 //
202 //
203 //
204 //
205 //
206 //
207 //
208 //
209 //
210 //
211 //
212 //
213 //
214 //
215 //
216 //
217 //
218 //
219 //
220 //
221 //
222 //
223 //
224 //
225 //
226 //
227 //
228 //
229 //
230 //
231 //
232 //
233 //
234 //
235 //
236 //
237 //
238 //
239 //
240 //
241 //
242 //
243 //
244 //
245 //
246 //
247 //
248 //
249 //
250 //
251 //
252 //
253 //
254 //
255 //
256 //
257 //
258 //
259 //
260 //
261 //
262 //
263 //
264 //
265 //
266 //
267 //
268 //
269 //
270 //
271 //
272 //
273 //
274 //
275 //
276 //
277 //
278 //
279 //
280 //
281 //
282 //
283 //
284 //
285 //
286 //
287 //
288 //
289 //
290 //
291 //
292 //
293 //
294 //
295 //
296 //
297 //
298 //
299 //
300 //
301 //
302 //
303 //
304 //
305 //
306 //
307 //
308 //
309 //
310 //
311 //
312 //
313 //
314 //
315 //
316 //
317 //
318 //
319 //
320 //
321 //
322 //
323 //
324 //
325 //
326 //
327 //
328 //
329 //
330 //
331 //
332 //
333 //
334 //
335 //
336 //
337 //
338 //
339 //
340 //
341 //
342 //
343 //
344 //
345 //
346 //
347 //
348 //
349 //
350 //
351 //
352 //
353 //
354 //
355 //
356 //
357 //
358 //
359 //
360 //
361 //
362 //
363 //
364 //
365 //
366 //
367 //
368 //
369 //
370 //
371 //
372 //
373 //
374 //
375 //
376 //
377 //
378 //
379 //
380 //
381 //
382 //
383 //
384 //
385 //
386 //
387 //
388 //
389 //
390 //
391 //
392 //
393 //
394 //
395 //
396 //
397 //
398 //
399 //
400 //
401 //
402 //
403 //
404 //
405 //
406 //
407 //
408 //
409 //
410 //
411 //
412 //
413 //
414 //
415 //
416 //
417 //
418 //
419 //
420 //
421 //
422 //
423 //
424 //
425 //
426 //
427 //
428 //
429 //
430 //
431 //
432 //
433 //
434 //
435 //
436 //
437 //
438 //
439 //
440 //
441 //
442 //
443 //
444 //
445 //
446 //
447 //
448 //
449 //
450 //
451 //
452 //
453 //
454 //
455 //
456 //
457 //
458 //
459 //
460 //
461 //
462 //
463 //
464 //
465 //
466 //
467 //
468 //
469 //
470 //
471 //
472 //
473 //
474 //
475 //
476 //
477 //
478 //
479 //
480 //
481 //
482 //
483 //
484 //
485 //
486 //
487 //
488 //
489 //
490 //
491 //
492 //
493 //
494 //
495 //
496 //
497 //
498 //
499 //
500 //
501 //
502 //
503 //
504 //
505 //
506 //
507 //
508 //
509 //
510 //
511 //
512 //
513 //
514 //
515 //
516 //
517 //
518 //
519 //
520 //
521 //
522 //
523 //
524 //
525 //
526 //
527 //
528 //
529 //
530 //
531 //
532 //
533 //
534 //
535 //
536 //
537 //
538 //
539 //
540 //
541 //
542 //
543 //
544 //
545 //
546 //
547 //
548 //
549 //
550 //
551 //
552 //
553 //
554 //
555 //
556 //
557 //
558 //
559 //
560 //
561 //
562 //
563 //
564 //
565 //
566 //
567 //
568 //
569 //
570 //
571 //
572 //
573 //
574 //
575 //
576 //
577 //
578 //
579 //
580 //
581 //
582 //
583 //
584 //
585 //
586 //
587 //
588 //
589 //
590 //
591 //
592 //
593 //
594 //
595 //
596 //
597 //
598 //
599 //
600 //
601 //
602 //
603 //
604 //
605 //
606 //
607 //
608 //
609 //
610 //
611 //
612 //
613 //
614 //
615 //
616 //
617 //
618 //
619 //
620 //
621 //
622 //
623 //
624 //
625 //
626 //
627 //
628 //
629 //
630 //
631 //
632 //
633 //
634 //
635 //
636 //
637 //
638 //
639 //
640 //
641 //
642 //
643 //
644 //
645 //
646 //
647 //
648 //
649 //
650 //
651 //
652 //
653 //
654 //
655 //
656 //
657 //
658 //
659 //
660 //
661 //
662 //
663 //
664 //
665 //
666 //
667 //
668 //
669 //
670 //
671 //
672 //
673 //
674 //
675 //
676 //
677 //
678 //
679 //
680 //
681 //
682 //
683 //
684 //
685 //
686 //
687 //
688 //
689 //
690 //
691 //
692 //
693 //
694 //
695 //
696 //
697 //
698 //
699 //
700 //
701 //
702 //
703 //
704 //
705 //
706 //
707 //
708 //
709 //
710 //
711 //
712 //
713 //
714 //
715 //
716 //
717 //
718 //
719 //
720 //
721 //
722 //
723 //
724 //
725 //
726 //
727 //
728 //
729 //
730 //
731 //
732 //
733 //
734 //
735 //
736 //
737 //
738 //
739 //
740 //
741 //
742 //
743 //
744 //
745 //
746 //
747 //
748 //
749 //
750 //
751 //
752 //
753 //
754 //
755 //
756 //
757 //
758 //
759 //
760 //
761 //
762 //
763 //
764 //
765 //
766 //
767 //
768 //
769 //
770 //
771 //
772 //
773 //
774 //
775 //
776 //
777 //
778 //
779 //
780 //
781 //
782 //
783 //
784 //
785 //
786 //
787 //
788 //
789 //
790 //
791 //
792 //
793 //
794 //
795 //
796 //
797 //
798 //
799 //
800 //
801 //
802 //
803 //
804 //
805 //
806 //
807 //
808 //
809 //

```

**OR**

# research project

groups of 3

pick a subject (list will be available)

## design & analysis

## experimentation



# Research Project: open questions

*LSM tree* file allocation *on Zoned SSDs*

flexible *Zoned SSDs* allocation scheme

exploit *near-sorted data* for join processing

*bitmap indexing* on complex SQL queries

come up with your **own topic!**

*more on the website (soon)*



# A good project

- (1) has a clear plan by project proposal by **end-February** (5%)
- (2) has significant preliminary work done by **end-March** (10%)

evaluation at the **end of the semester** (25%)

- (i) present the key ideas of the implementation/new approach
- (ii) present a set of experiments supporting your claims

come to OH!

(more details for the projects in Class 4)





# Class Goal

understand the internals of  
data systems for data science

tune data systems through **adaptation** and **automation**

get acquainted with research in the area

# Can I take this class?



## **background**

C++ programming

data structures

algorithms

comp. architecture

## **pre-req**

CS460/660 & CS210

contact Manos if not sure

## **how to be sure?**

if familiar with most, then maybe!

if familiar with **none**, then no!

# Next classes

## **Class 1-2**

logistics, big data, data systems, trends and outlook

## **Class 3**

more basics on data systems, systems classification, graph, cloud

## **Class 4**

intro to class project

## **Class 5 and beyond**

present and **discuss** research papers from Manos + students + guest lectures

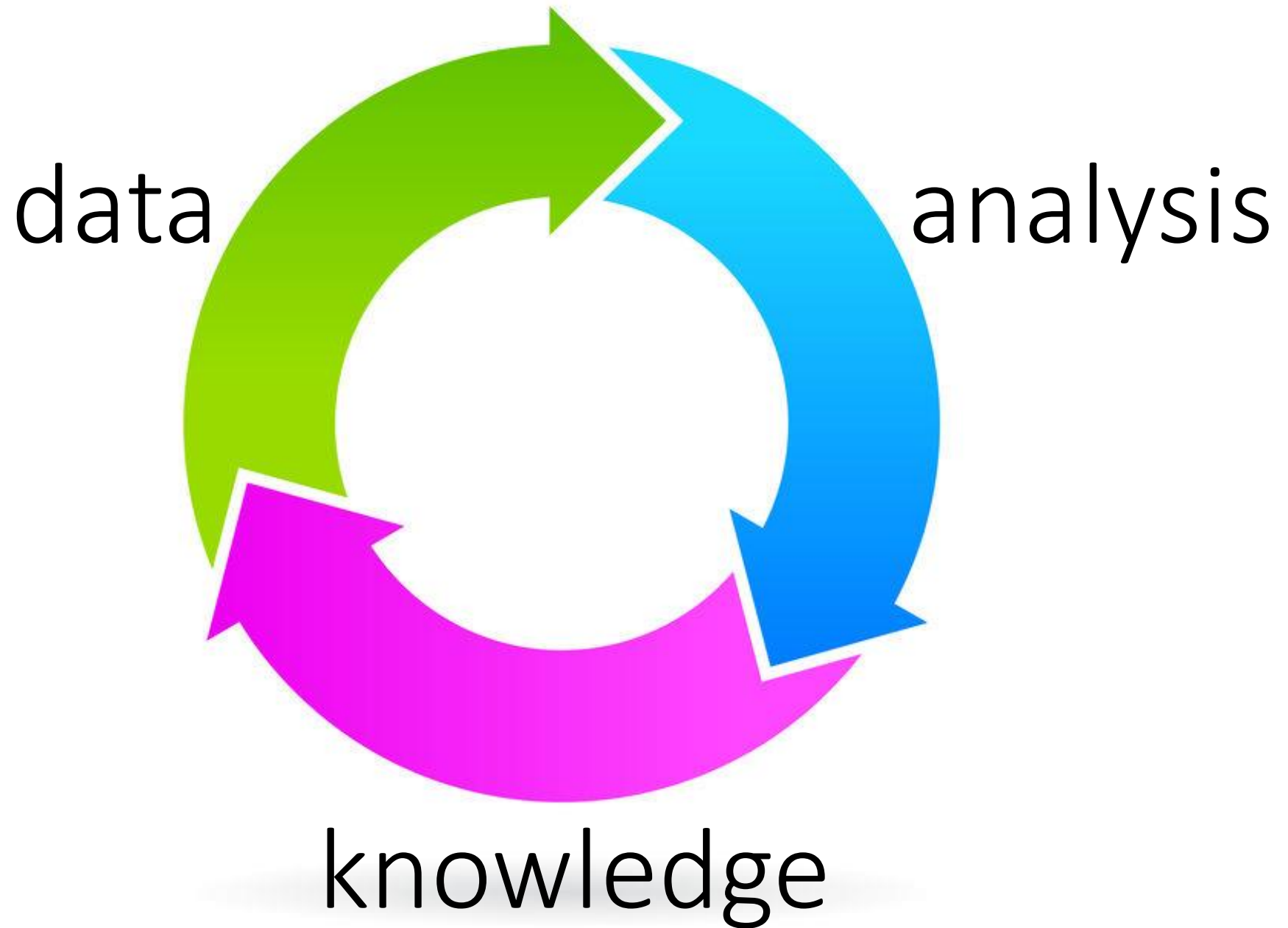


*who doesn't have a lot of data?*



big data?

So what do we do with this data?

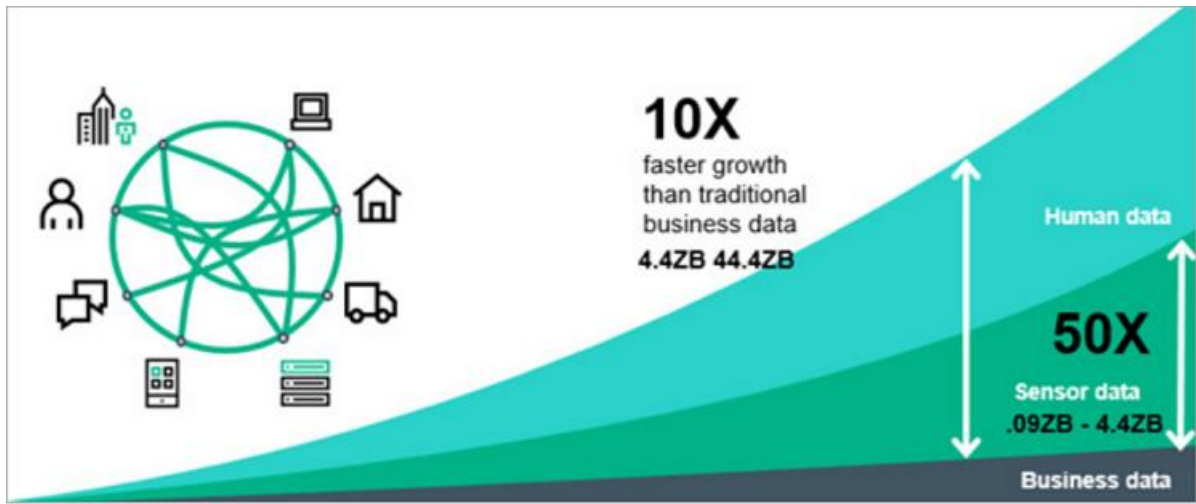


is data  
analysis new?

? what is  
*really* new?



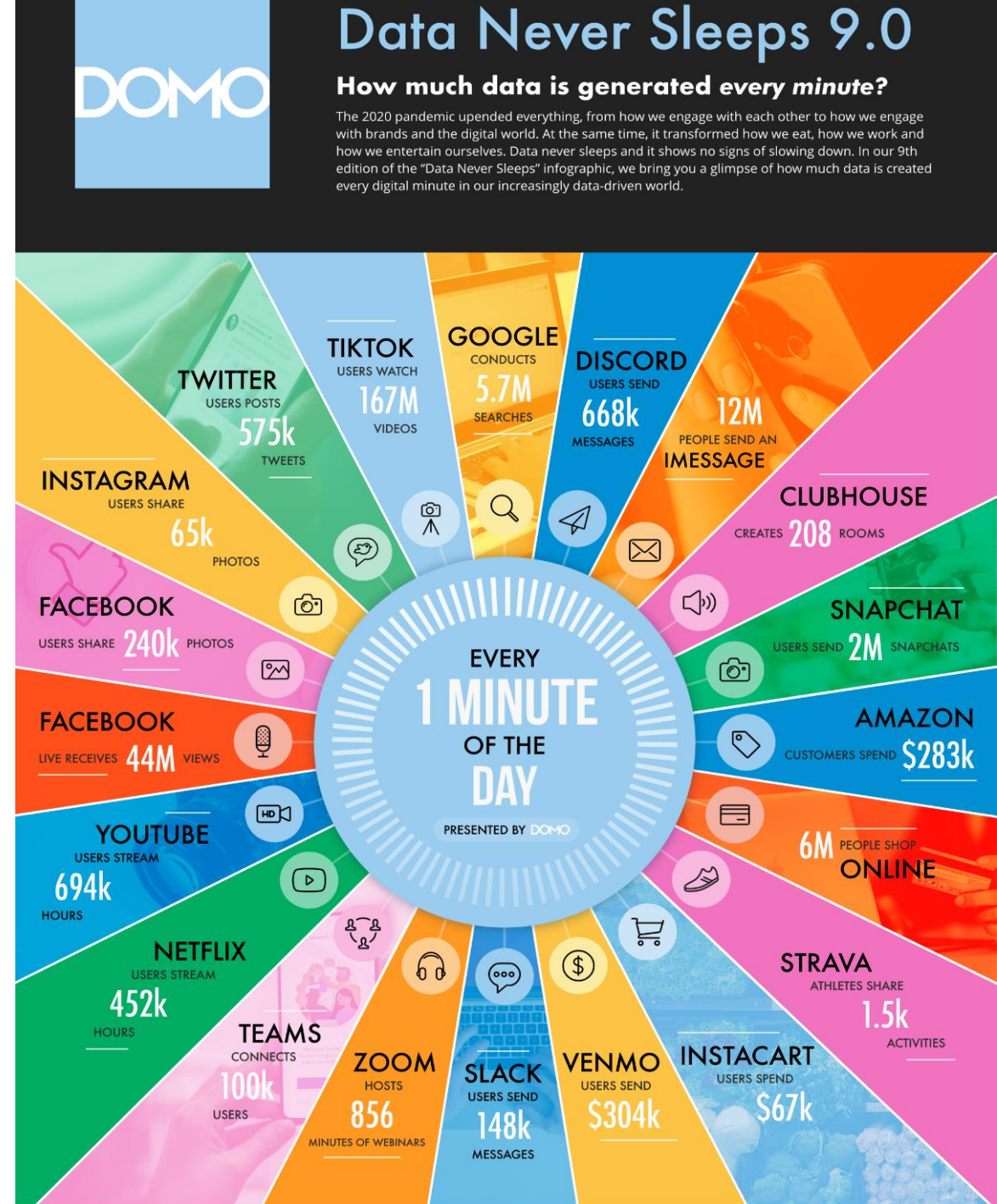




Every day, we create 2.5 exabytes\* of data — 90% of the data in the world today has been created in the last two years alone.

[Understanding Big Data, IBM]

\*exabyte =  $10^9$  GB



# data management skills needed



100s of entries

**pen & paper**

$10^3$ - $10^6$  of entries

**UNIX tools and excel**

$10^9$  of entries

**custom solutions, programming**

$10^{12+}$  of entries

**data systems**



size (volume)

rate (velocity)

sources (variety)




big data

(it's not only about size)

*all of the above plus ...*


our ability to collect *machine-generated* data

 scientific experiments

 sensors

social 

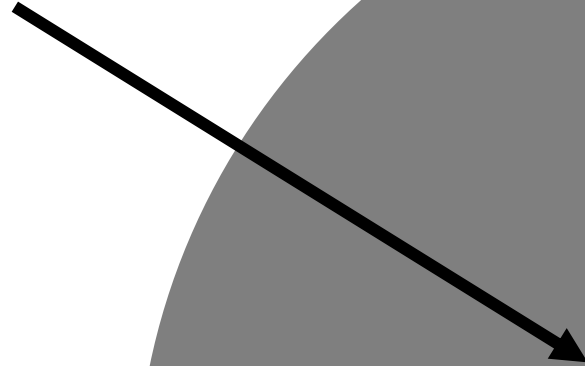
monitoring 

 micro-payments

Internet-of-Things 

cloud 

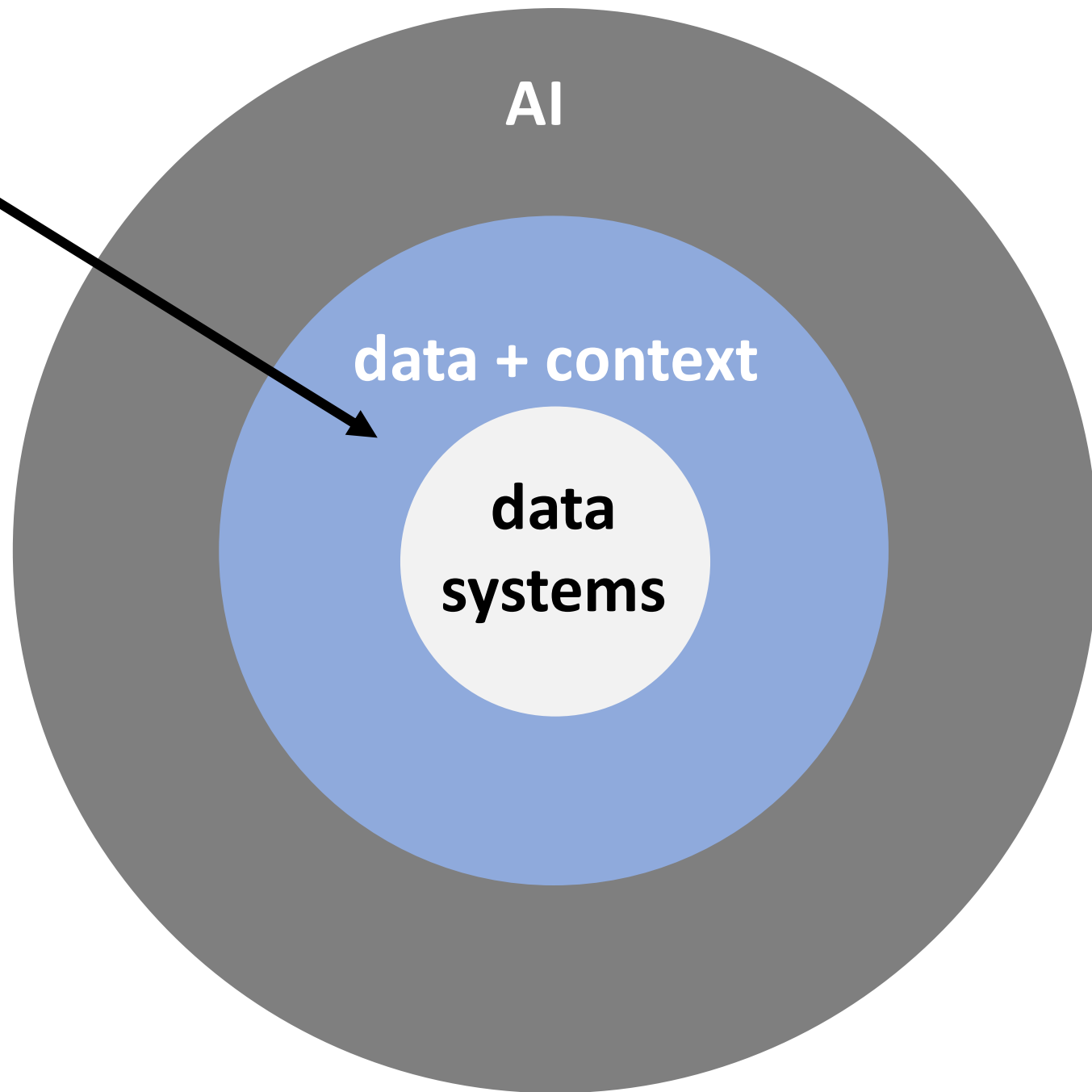
data systems are  
in the middle of this!



**big data**

**data  
systems**

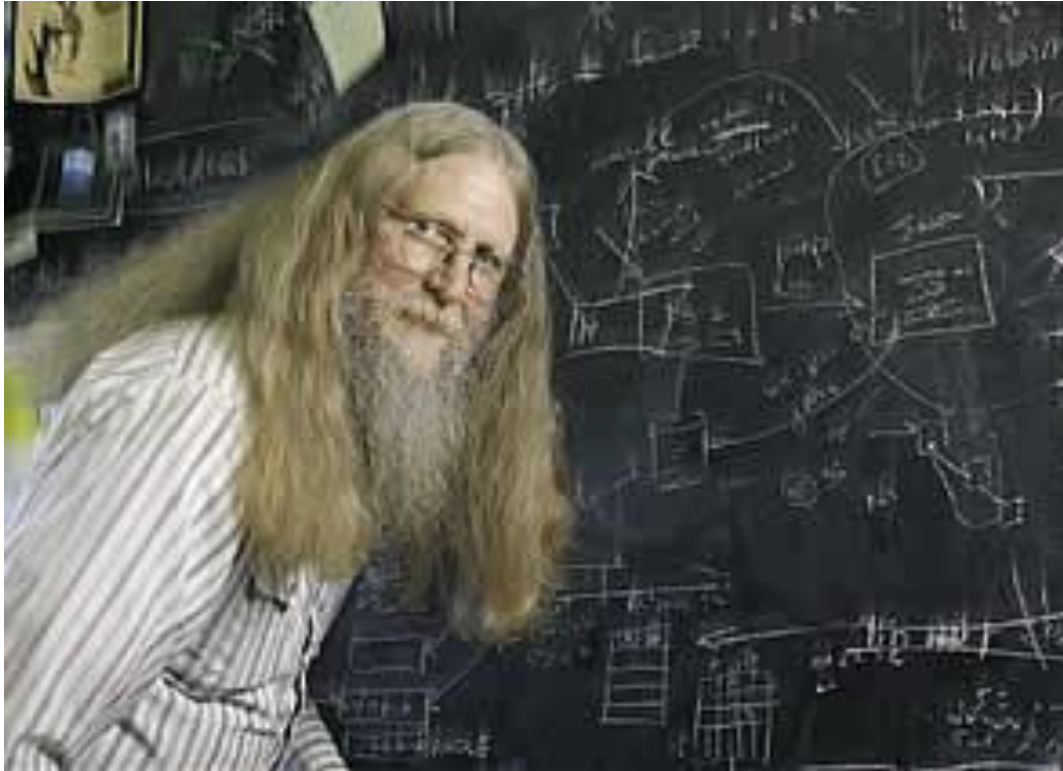
data systems are  
in the middle of this!



what is a **data system**?

a **data system** is a large software system  
(a collection of algorithms and data structures)  
that **stores data**, and provides the **interface** to  
**update** and **access** them **efficiently**

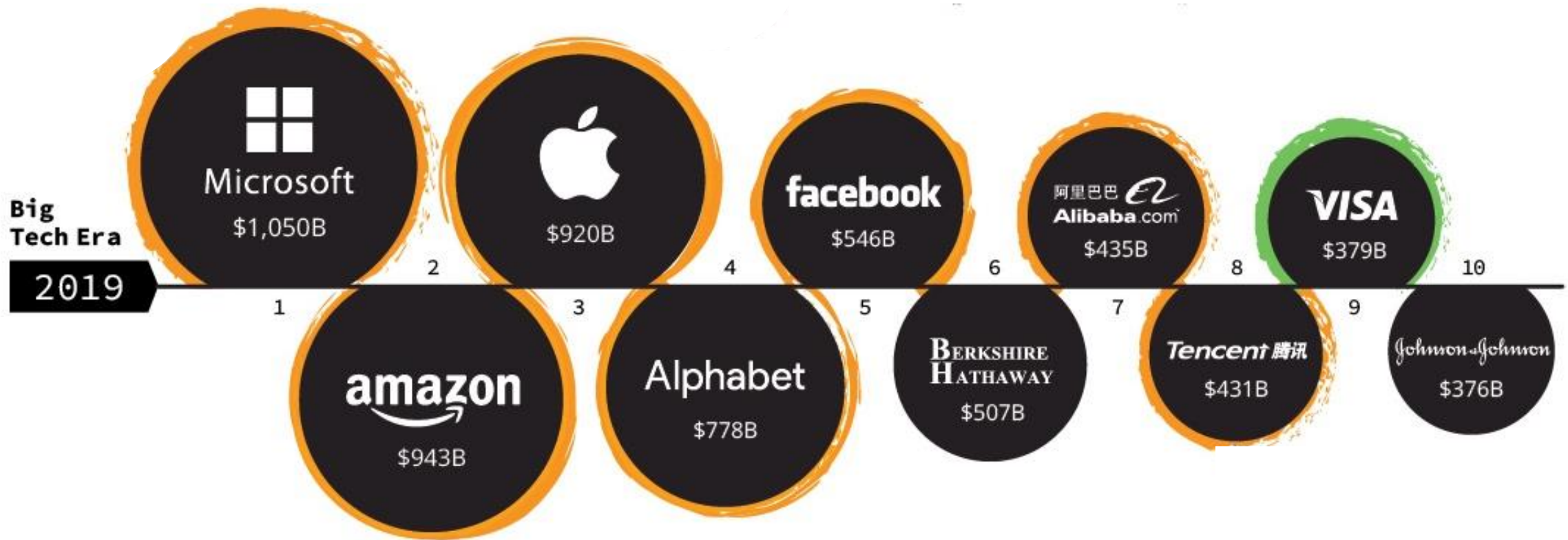
the end goal is to make **data analysis** easy



*“relational databases  
are the foundation of  
western civilization”*

Bruce Lindsay, IBM Research

ACM SIGMOD Edgar F. Codd Innovations award 2012



***+ growing need for tailored systems***



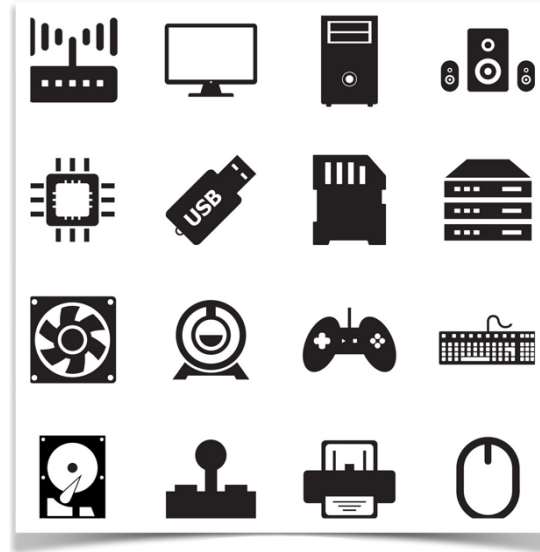
# Why?



more data



new hardware



new applications



new performance  
goals



# The big success of 6 decades of research

a declarative interface!

ask ***what*** you want

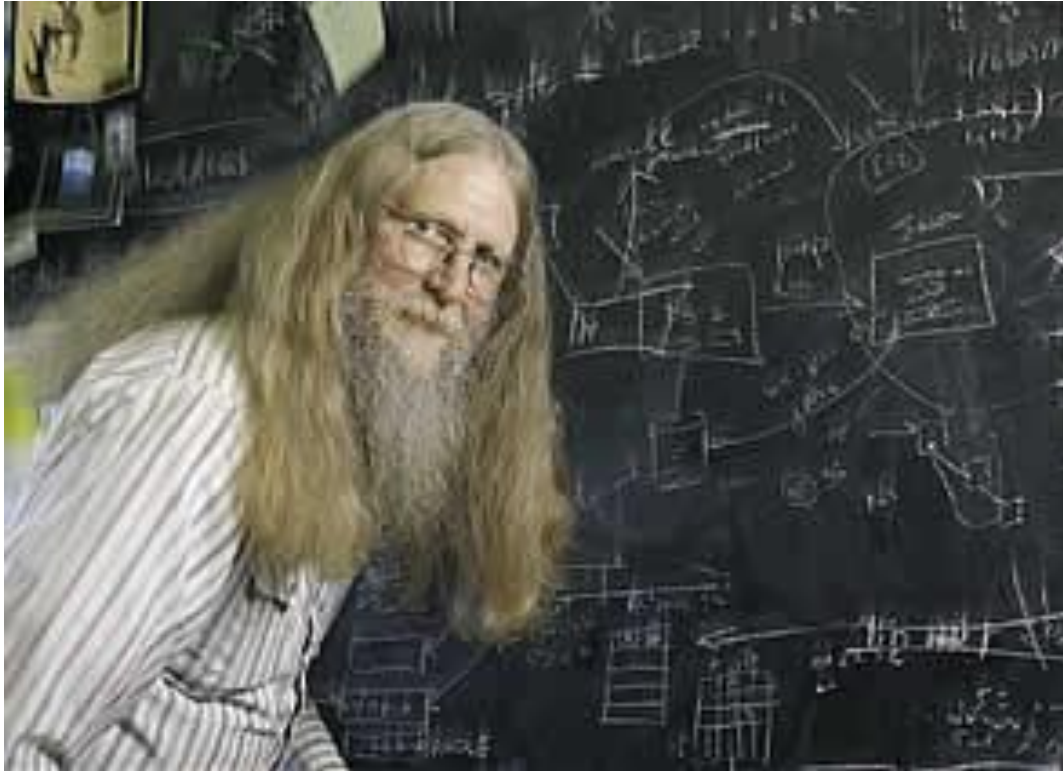
“ask and thou shall receive”

data system

system decides ***how***  
*to store & access*



is this good?

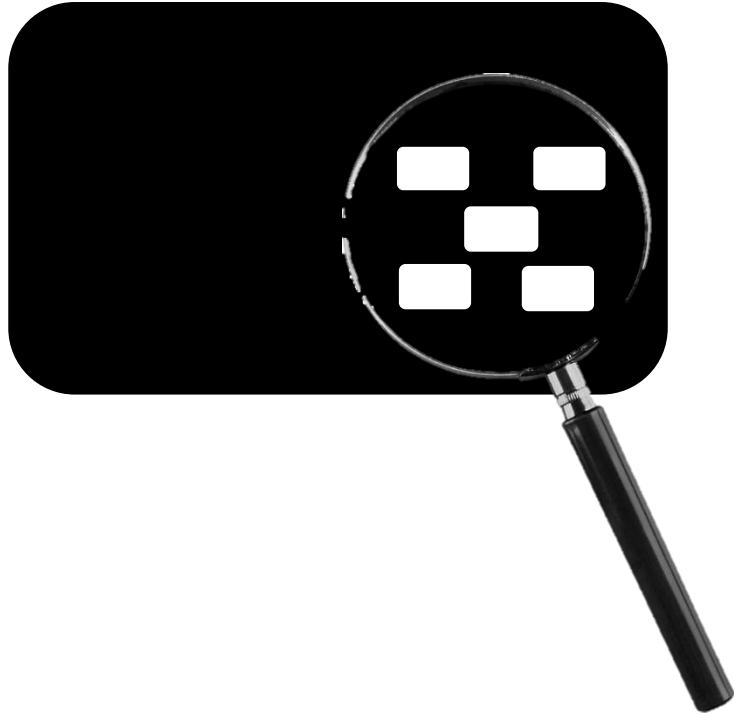


*“three things are important  
in the database world:  
**performance, performance,  
and performance**”*

Bruce Lindsay, IBM Research

ACM SIGMOD Edgar F. Codd Innovations award 2012

# CS561: data systems **kernel** under the looking glass



this is where we will spend our time!

system architecture (row/column/hybrid)

indexing

relational/graph/key-value

scale-up/scale-out

***goal: learn to design and implement a DB kernel***

# how to design a data system kernel?

what are its basic components?

algorithms/data structures/caching policies

what decisions should we make?

how to combine? how to optimize for hardware?

*designing a DB kernel is **complex***

# data system design complexity



application



performance



budget

thousands of options  
millions of decisions  
billions of combinations

# let's think together: a simple DB kernel

a key-value system, each entry is a {key,value} pair

**main operations:** *put, get, scan, range scan, count*

workload has both reads (*get, scan, range scan*) *and writes (put)*

data

how to store and how to access data?

how to efficiently delete?



# designing a simple key-value system

what is the key/value?

are they stored together?

can read/write ratio change over time?

what to use? b-tree, hash-table, scans, skip-lists, zonemaps?

how to handle concurrent queries? million concurrent queries?

what happens if data does not fit in memory?

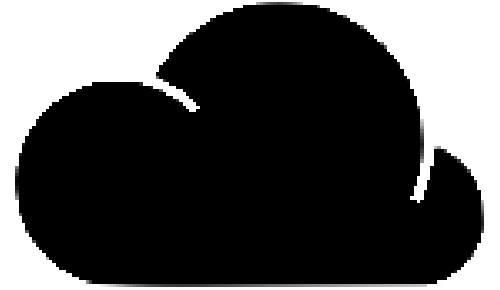
how to compress data?

what about privacy and security?

how to offer robustness guarantees?



# what happens when we move to the cloud?



hardware at massive scale

performance tradeoffs different

10GB app: 1% less memory in your machine

so what?

10GB app: 1% less memory in 1M instances

$1M * 10GB * 1\% = 100TB!$

~800k\$ in today's price

class key goal

understand **system design tradeoffs**

**design** and **prototype** a system

with other **side-effects**:

**sharpening your systems skills**

**(C/C++, profiling, debugging, linux tools)**

data system designer & researcher  
any business, any startup, any scientific domain

# CS 561: more logistics

## **topics**

storage layouts, HTAP systems, adaptive indexing, solid-state storage, data integration, data skipping, data systems and ML, learned index

## **past but still relevant topics**

relational systems, row-stores, query optimization, concurrency control, SQL

**no textbook – only research papers**

# grading



class participation: 5%

quizzes: 10%

project 0: 5%

project 1: 10%

paper presentation/student discussion: 15%

project proposal: 5%

mid-semester project report: 10%

project: 25%

exam: 15%

# Survival Guide

class website: <https://bu-disc.github.io/CS561/>

## Project 0 [5%]

- Individual
- Due on **Jan 31**

## Project 1 [10%]

- 3 persons per group
- Due on **Feb 14**

## Paper Presentation/Student Discussion [15%]

- Groups of ~10 in subgroups: presenters/critics/proponents
- Signup soon here: <http://tinyurl.com/S26-CS561-presentations>

## Quizzes on Technical Questions [10%]

- 12 individual in-class Quizzes
- Lowest two scores will be discarded

## Class Participation [5%]

## Class Project [40%]

- 2-3 persons per group
- Project proposal (5%), due on **Feb 23**
- Mid-way Report (10%), due on **Mar 22**
- Final Report + Presentation + Contribution (25%), due on **Apr 26**

## Exam [15%]

# Piazza



all discussions & announcements

<https://piazza.com/bu/spring2026/cs561>

also available on class website

no   
smartphones

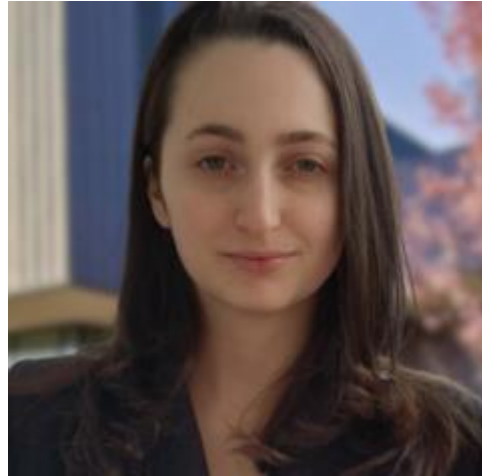
no   
laptop

**Why?**

there is enough evidence that laptops and phones slow you down



# Your awesome class staff!



Teona  
(Head TF)



Can

**Prof. Manos Athanassoulis**  
name in greek: Μάνος Αθανασούλης

grew up in Greece  
enjoys playing basketball and the sea

**BSc and MSc @ University of Athens, Greece**  
**PhD @ EPFL, Switzerland**  
**Research Intern @ IBM Research Watson, NY**  
**Postdoc @ Harvard University**

**some awards:**

NSF CAREER Award  
Best of VLDB/SIGMOD  
Facebook Faculty Fellowship



photo for VISA / conferences



Myrtos, Kefalonia, Greece

<http://cs-people.bu.edu/mathan/>

Office: CDS928

Office Hours: (see Piazza)

# How can I prepare?

## 1) Read background research material

- **Architecture of a Database System.**  
By J. Hellerstein, M. Stonebraker and J. Hamilton.  
Foundations and Trends in Databases, 2007
- **The Design and Implementation of Modern Column-store Database Systems.**  
By D. Abadi, P. Boncz, S. Harizopoulos, S. Idreos, S. Madden.  
Foundations and Trends in Databases, 2013
- **Data Structures for Data-Intensive Applications: Tradeoffs and Design Guidelines.**  
By M. Athanassoulis, S. Idreos, D. Shasha.  
Foundations and Trends in Databases, 2024

## 2) Start going over the papers

# class summary

2 classes + 5 OH + 1 Lab (5 days) per week

Quizzes with Technical Questions

Paper Presentation/Student Discussion

project 0 + project 1 + ***systems or research project***



proposal + mid-semester report + final report + project presentation

# what to do now?

- A) read the syllabus and the website**
- B) register to Piazza + Gradescope**
- C) start working on project 0**
- D) register for the presentation (week 2-3)
- E) start reading papers & prepare for tech. questions (week 3)
- F) go over the class project (end of next week will be available)
- G) start working on the proposal (week 3)

# Resources

**class website:** <https://bu-disc.github.io/CS561/>

**piazza website:** <https://piazza.com/bu/spring2026/cs561>

**presentation registration:** <http://tinyurl.com/S26-CS561-presentations>

**gradescope:** <https://www.gradescope.com/courses/1202939> (**code in Piazza**)

**office hours:** <https://piazza.com/bu/spring2026/cs561/staff>

**material:** papers available from the BU network

# Welcome to CS 561: Data Systems Architectures!

Prof. Manos Athanassoulis

[mathan@bu.edu](mailto:mathan@bu.edu)

next time: more detailed logistics and start with data systems design