COSI 167A

Advanced Data Systems

Class 1

Welcome to COSI 167A!

Prof. Subhadeep Sarkar

https://ssd-brandeis.github.io/COSI-167A/

Why take the class?

Introduction to “modern” databases!

BIG data — **Data-driven world, Unstructured data**

store and **manage** data — Data is generated at an **unprecedented rate** and **volume**
—**“Does your system SCALE?”**

querying BIG data & querying **fast** — **Querying unstructured data, SQL?**

new **system designs** — **New application requirements = New design trends**

getting your **hands “dirty”** — **Play with large-scale, commercial storage engines**

The **first rule** of class

Ask!

Ask questions!

& answer my questions!

The **first rule** of class

Ask!

Ask questions!

& answer my questions!

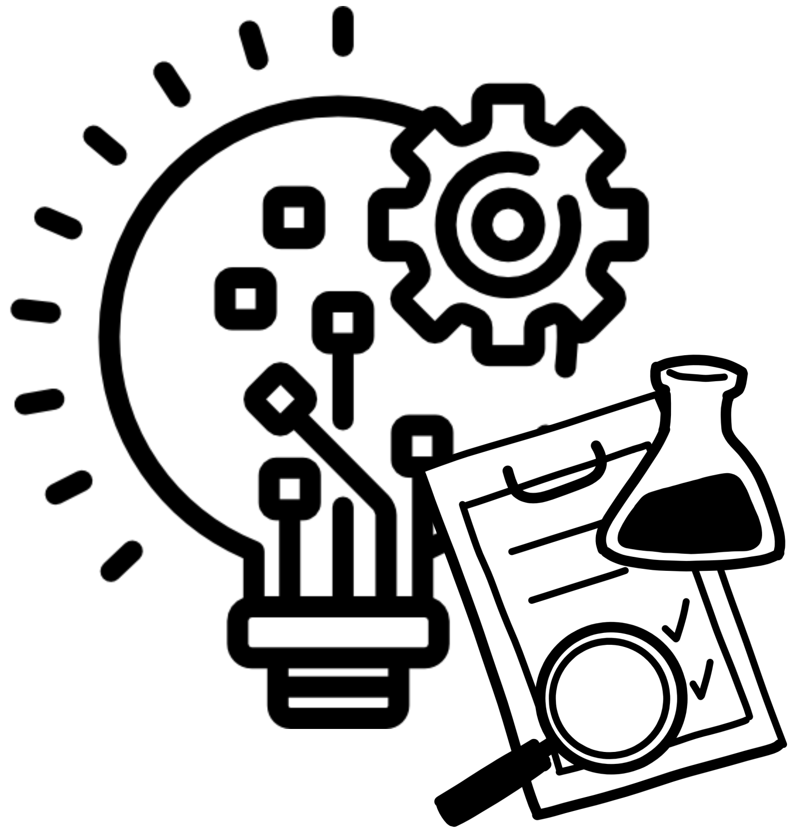
advanced topics

foster discussion

There’s NO stupid question!

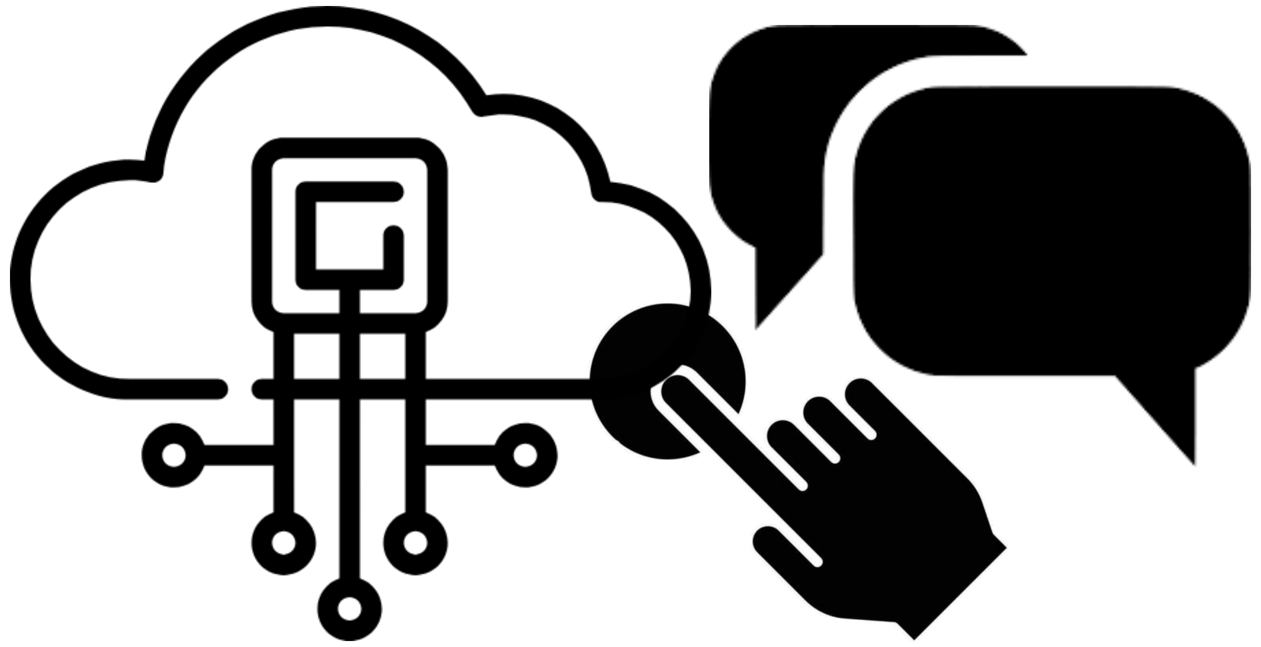
Class **Philosophy**

Principles to go by



cutting-edge research

question everything



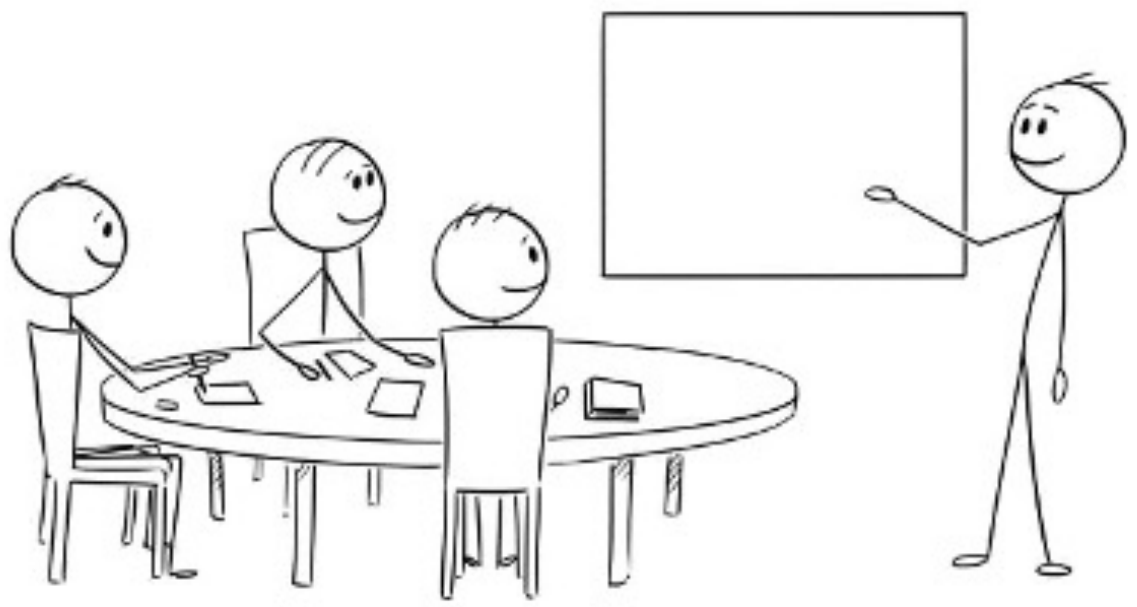
interactive & collaborative

Learn to *think out of the box*!

What do we do in this class?

Class components

reading papers



presentations

reviews &
technical questions

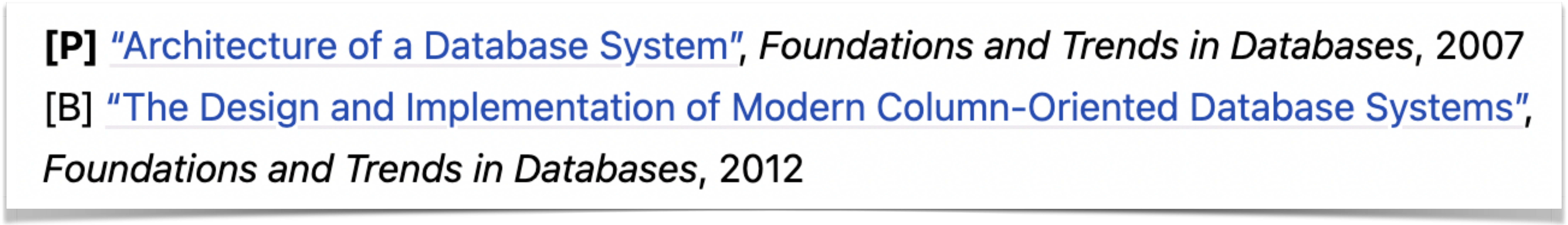
projects

Reading papers

Getting familiarized with the state of the art

Discuss in detail **1-2 research papers** every class

papers with **[P]** are the ones **presented**; [B] indicates **background papers**



[P] "Architecture of a Database System", *Foundations and Trends in Databases*, 2007

[B] "The Design and Implementation of Modern Column-Oriented Database Systems", *Foundations and Trends in Databases*, 2012

Reading papers

Getting familiarized with the state of the art

Discuss in detail **1-2 research papers** every class

papers with **[P]** are the ones **presented**; [B] indicates **background papers**

Read 'em all, and try to acquire the following skills.

learn to **read technical papers**

learn to **critique constructively**

learn to **prepare slides** & **present**

Brandeis UNIVERSITY

Reading papers

Getting familiarized with the state of the art

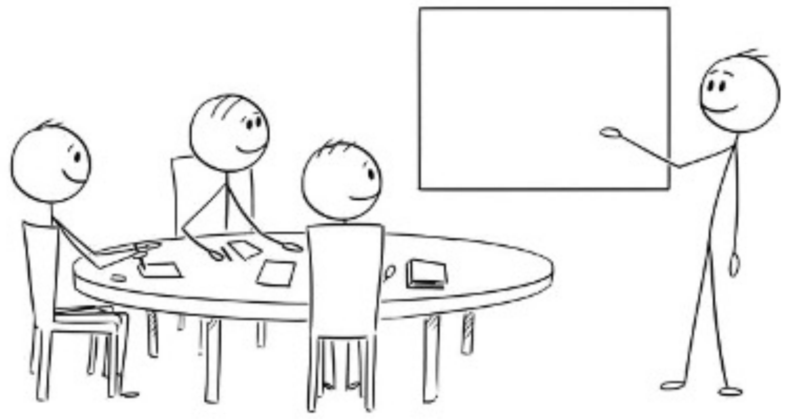
Discuss in detail **1-2 research papers** every class

papers with **[P]** are the ones **presented**; [B] indicates **background papers**

Read ’em all!

Discussion format: **Lectures**
Guest lectures
Student presentations

Write **paper reviews** and answer **technical questions**



Presenting papers

Mastering the art of presentation

2 students will be responsible for presenting the paper

learn the art of **technical presentation**, think as: **visualizing a review**

during the presentation, **anyone can ask questions**

each question is **addressed to all** (including me!)

prepare slides **at least a week before** your presentation

get your **slides reviewed** by me **twice** before your final presentation

Projects

Let's get our hands dirty

Project 1 **Individual** project

Out **next week**; due in **two weeks**

Goal: Sharpen your **programming & system development** skills

May choose the programming **language of your choice**

[C/C++?]

More on Project 1 next week!

Projects

Let's get our hands dirty

Class Project

Groups of 2

Out **mid-September**; due in **December**

Multiple milestones in between

Systems project

Research project

Projects

Let's get our hands dirty

Class Project

Systems project

Research project

Projects

Let’s get our hands dirty

Class Project

Systems project

Research project

implementation-heavy project

develop, **experiment**, and **analyze**

Projects

Let's get our hands dirty

Class Project

Systems project

implementation-heavy project

develop, **experiment**, and **analyze**

Research project

pick a topic (list available *soon*)

design, **develop**, and **analyze**

large-scale experimentation

Class project: **Theme**

Working with state-of-the-art systems

NoSQL key-value stores

RocksDB

cassandra

APACHE HBASE

MEMCACHED

AsterixDB

Google BigTable

amazon DynamoDB

redis

A **good** class project

Plan ahead in time

has a clear plan by project proposal by **first week of October**

identify the **project requirements**, the **challenges**, & the **milestones** 5%

has significant preliminary work done by **first week of November**

each question is **addressed to all** (including me!) 5%

evaluation at the **end of the semester**

present the **key ideas** of the **implementation/new approach** 10%

present **experimental results** that support your claims 20%

Pro tip: Come to student hours!

GOOD JOB! WELL DONE GOOD JOB!

The **ultimate reward**

For a job well done!

GOOD JOB! WELL DONE GOOD JOB!

ACM SIGMOD Student Research Competition

ACM Special Interest Group in Data Management (**SIGMOD**)

top conference in **data management**

receives submissions of **student research**

top 10-15 are invited to present their work **at the conference**

top-3 projects get an award and invitation to present **at the ACM level**

The **ultimate reward**

For a job well done!

GOOD JOB!
WELL DONE
GOOD JOB!

Berlin, Germany

The **ultimate reward**

For a job well done!

GOOD JOB! WELL DONE GOOD JOB!

Santiago, Chile

Anatomy of LSM Memory Buffer: Insights & Implications

Shubham Kaushik

Subhadeep Sarkar

SSD

Brandeis UNIVERSITY

dbtest.io

Santiago, Chile 2024

Can I take the class?

Can I?

Background

programming

data structures and **algorithms**

databases fundamentals

Pre-requisite

COSI 127B or equivalent

up for some **challenge**

Still not sure? Contact Subhadeep

Data is **everywhere**!

Data is **everywhere**!

Everyone produces data!

experimental **physics** (IceCube, CERN)
neuroscience
biology

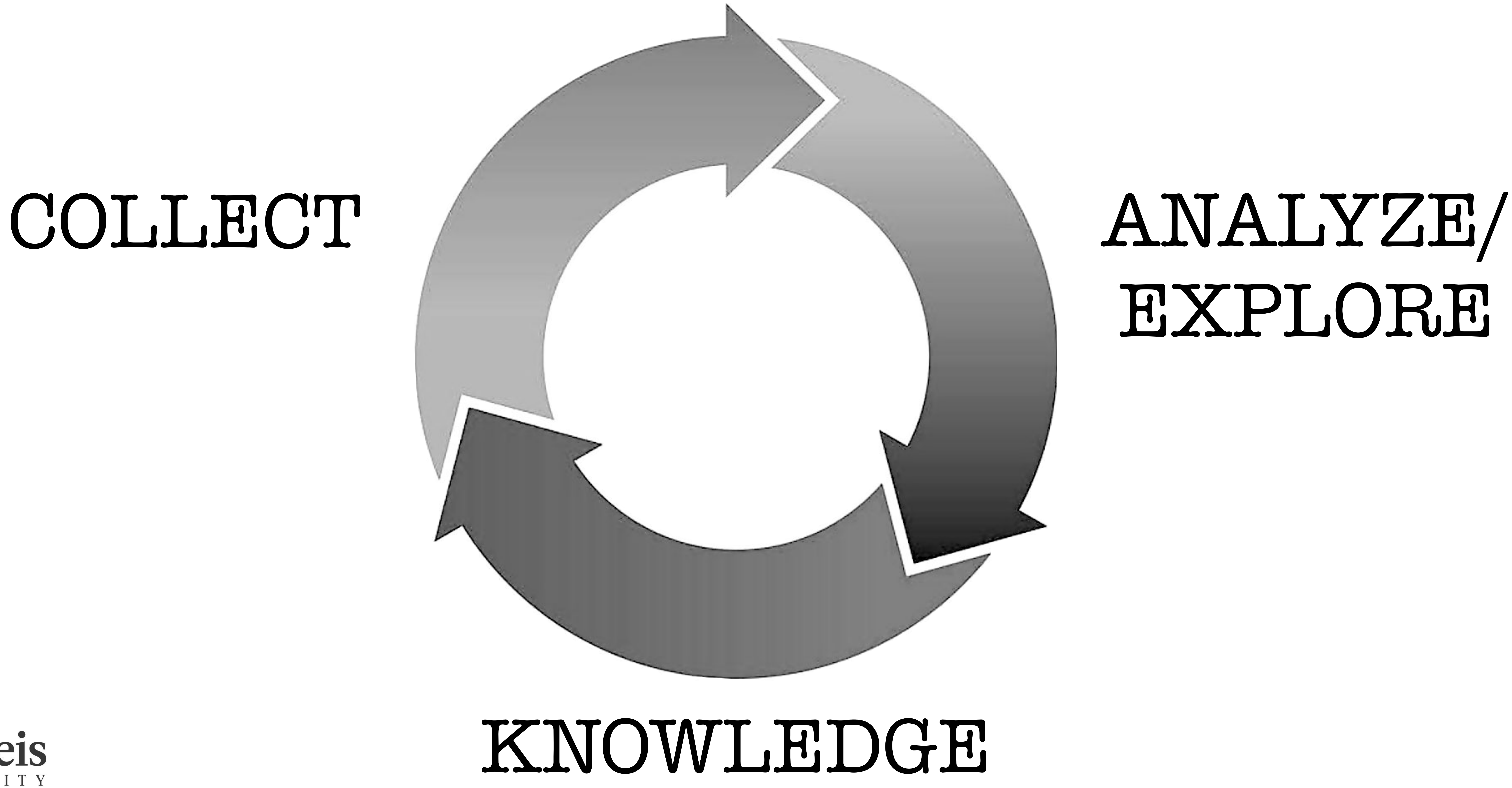
data mining business datasets
machine learning and AI for corporate and consumer

micro-transactions, economics

Brandeis UNIVERSITY

Data is **everywhere**!

Everyone produces data!



Thought Experiment 1

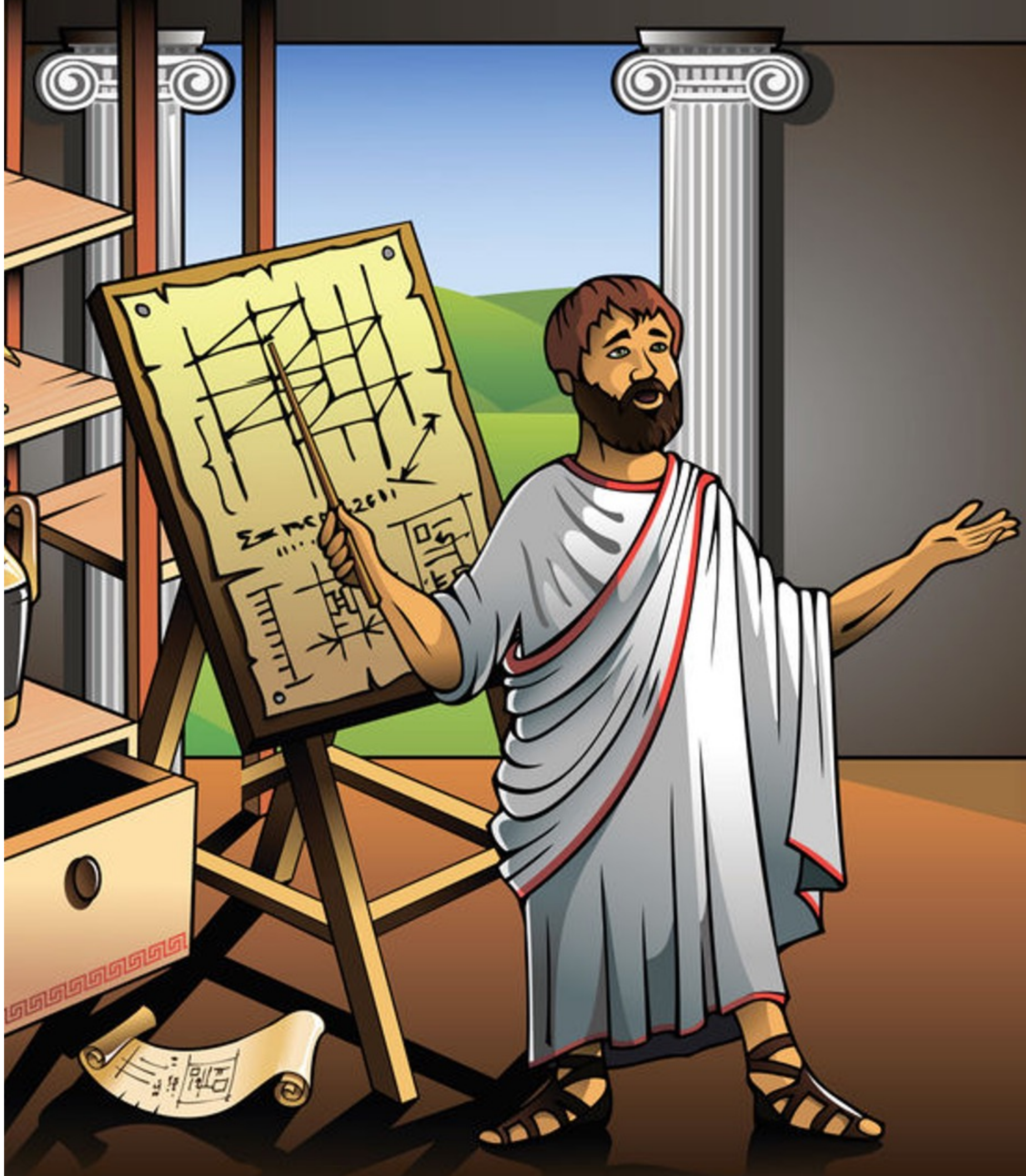
Is **data analysis** new?

NOT REALLY

Thought Experiment 2

What’s really new in modern **data analysis**?

SCALE!



Big data

How big is big?

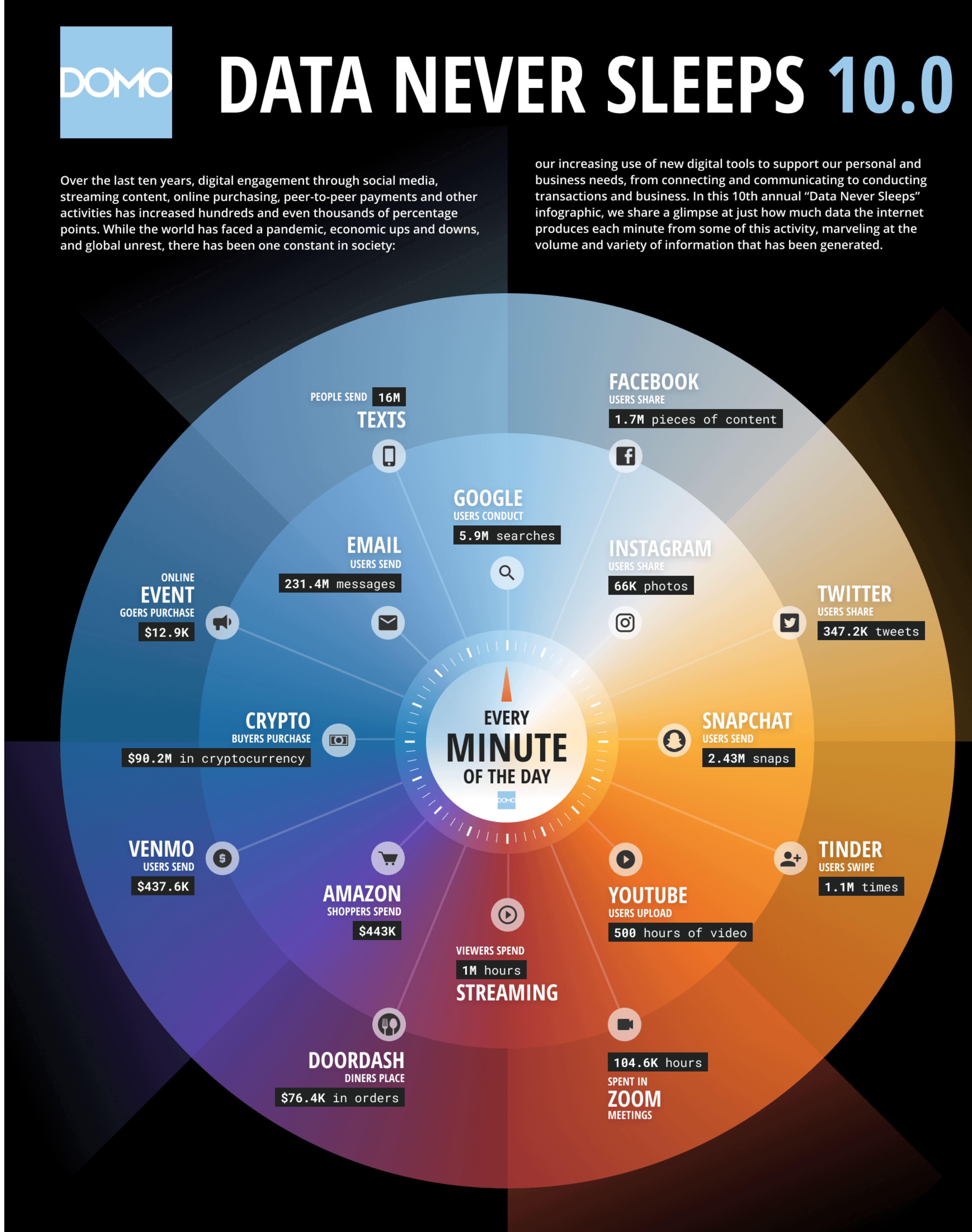
exponential data growth

efficient database systems

Big data

How big is big?

> *Every two days we generate as much data as we did since the dawn of humanity until 2003.*
>
> — **Eric Schmidt** (CEO, Google), 2010



DOMO **DATA NEVER SLEEPS 10.0**

Over the last ten years, digital engagement through social media, streaming content, online purchasing, peer-to-peer payments and other activities has increased hundreds and even thousands of percentage points. While the world has faced a pandemic, economic ups and downs, and global unrest, there has been one constant in society: our increasing use of new digital tools to support our personal and business needs, from connecting and communicating to conducting transactions and business. In this 10th annual “Data Never Sleeps” infographic, we share a glimpse at just how much data the internet produces each minute from some of this activity, marveling at the volume and variety of information that has been generated.

EVERY MINUTE OF THE DAY
PEOPLE SEND 16M TEXTS
FACEBOOK USERS SHARE 1.7M pieces of content
GOOGLE USERS CONDUCT 5.9M searches
EMAIL USERS SEND 231.4M messages
INSTAGRAM USERS SHARE 66K photos
ONLINE EVENT GOERS PURCHASE $12.9K
TWITTER USERS SHARE 347.2K tweets
CRYPTO BUYERS PURCHASE $90.2M in cryptocurrency
SNAPCHAT USERS SEND 2.43M snaps
VENMO USERS SEND $437.6K
TINDER USERS SWIPE 1.1M times
AMAZON SHOPPERS SPEND $443K
YOUTUBE USERS UPLOAD 500 hours of video
VIEWERS SPEND 1M hours STREAMING
DOORDASH DINERS PLACE $76.4K in orders
104.6K hours SPENT IN ZOOM MEETINGS
DOMO

Big data

How big is big?

> *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

DATA NEVER SLEEPS 1.0 VS. 10.0
3x
2M
5.9M
2013
2022
GOOGLE
USER QUERIES
10x
48
500
2013
2022
YOUTUBE
HOURS UPLOADED
18x
3.6K
66K
2013
2022
INSTAGRAM
PHOTOS SHARED
3.5x
100K
347K
2013
2022
TWITTER
TWEETS SHARED
2.5x
684K
1.7M
2013
2022
FACEBOOK
CONTENT SHARED
1.1x
204M
231M
2013
2022
EMAILS
EMAILS SENT

Big data

Is it only about size?

size (**v**olume)

rate (**v**elocity)

sources (**v**ariety)

5 V’s

accuracy (**v**eracity)

utility (**v**alue)

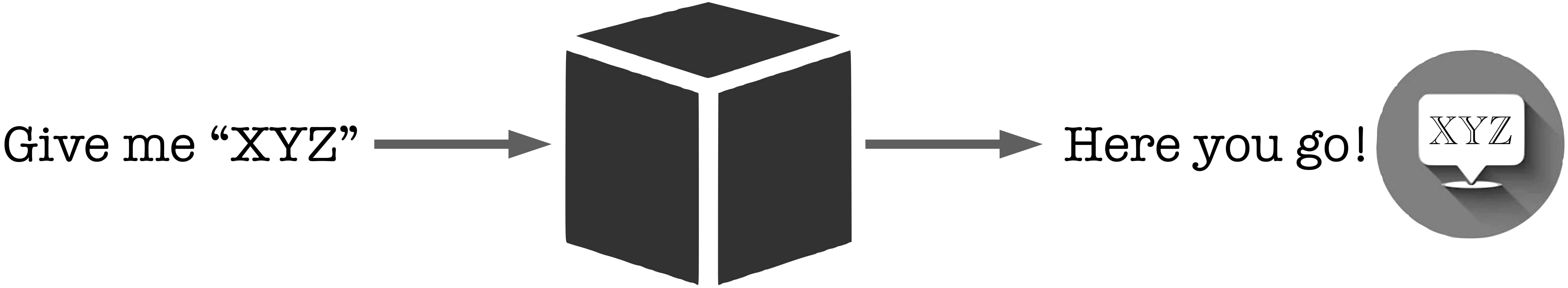
Managing big data

Operating at scale

100s of entries	**Pen & paper**
10^3-10^6 of entries	**UNIX tools and excel**
10^6-10^{12} of entries	**Custom solutions, programming**
$>10^{12}$ of entries	**Data systems**

Data systems

What are they, really?



Give me “XYZ”

Here you go!

declarative box

Data systems

What are they, really?

A data system is an **end-to-end software system** that is responsible for **storing data** and **providing access to the data** through **efficient data movement.**

Data systems

What are they, really?

A data system is an **end-to-end software system** that is responsible for **storing data** and **providing access to the data** through **efficient data movement.**

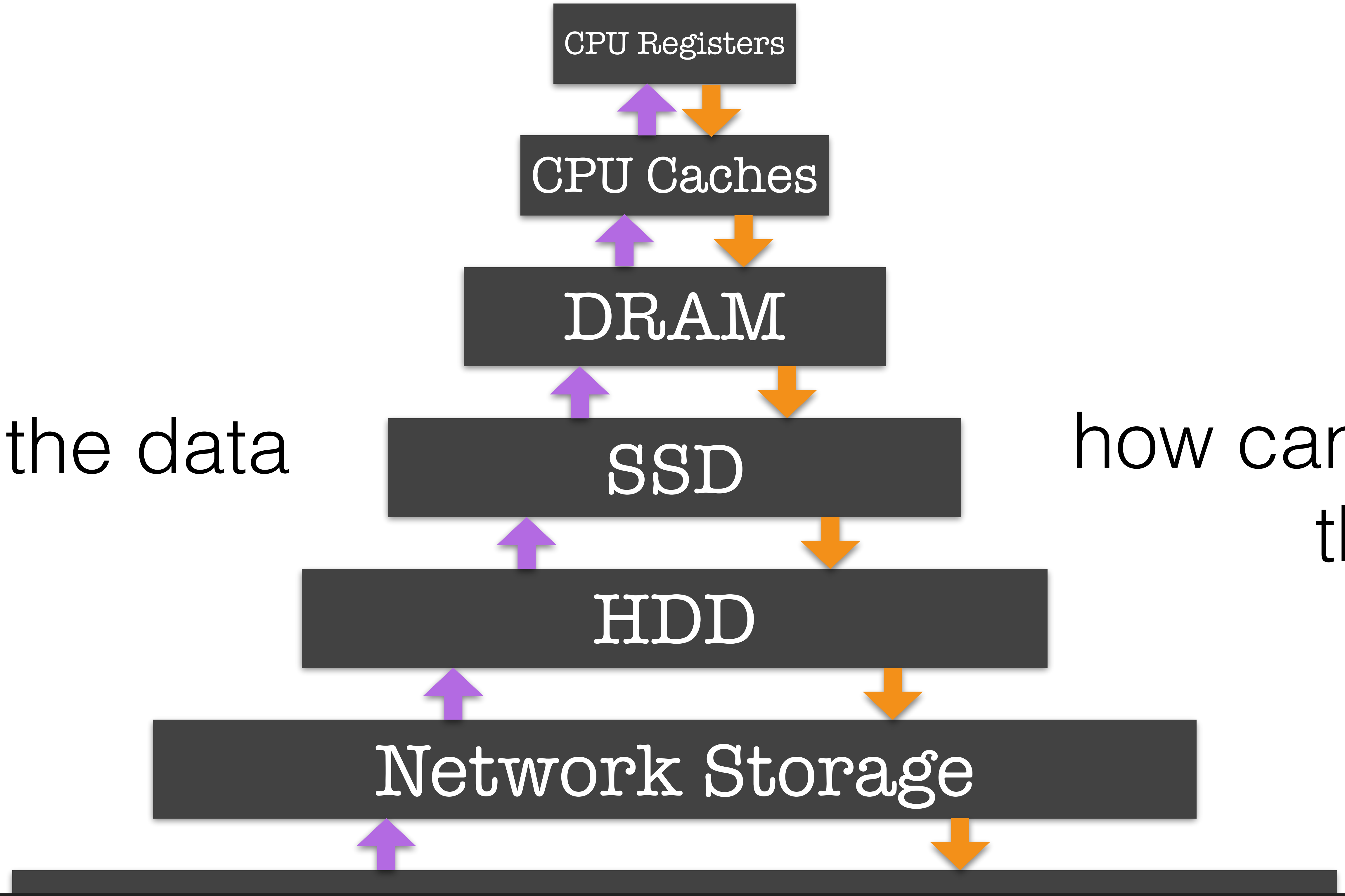
how can we **organize** the data in collections

how can we **process/analyze** the data quickly

Data systems

What are they, really?

how can we **organize** the data in collections

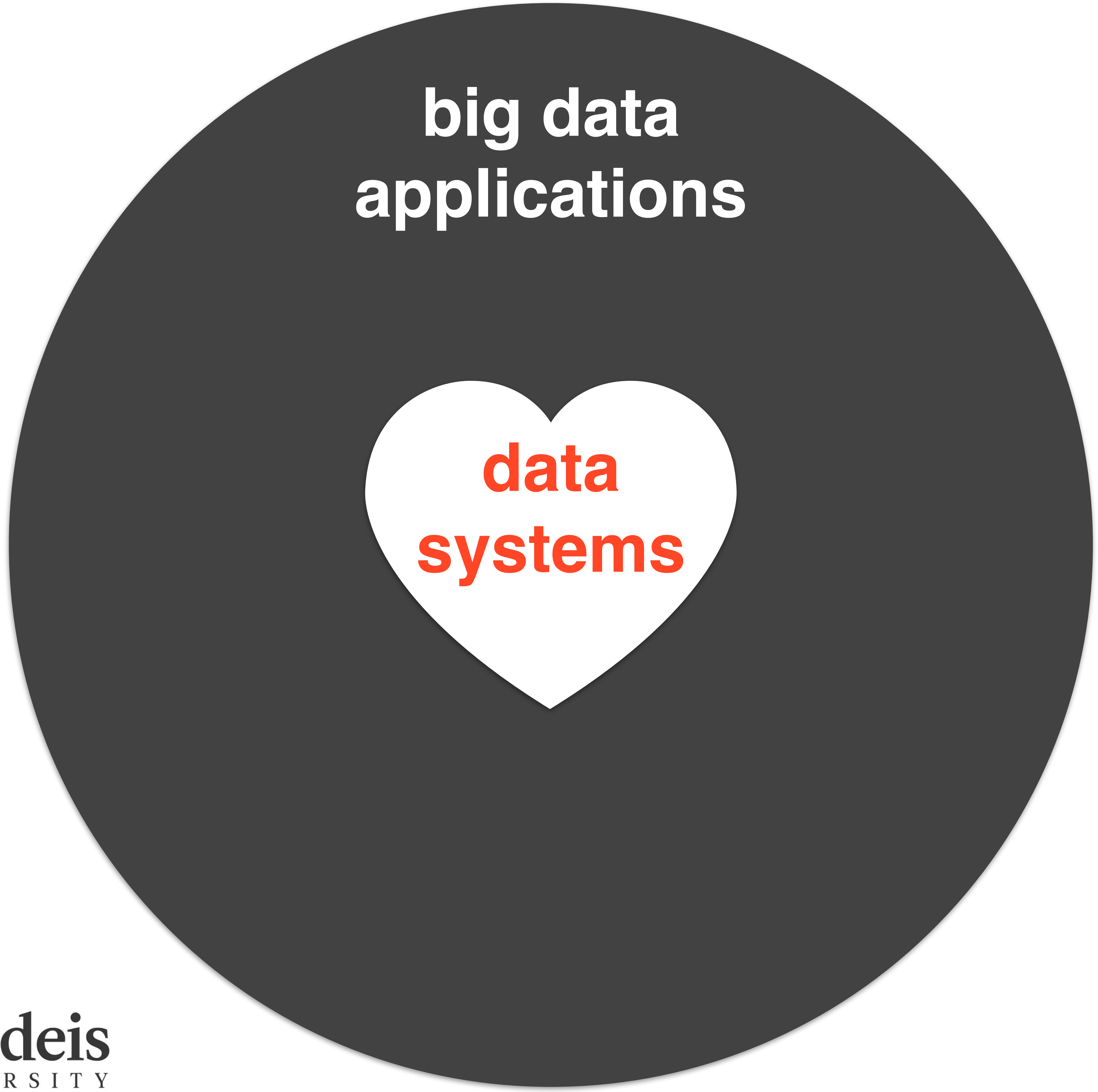


how can we **process/analyze** the data quickly

>70% of time is taken to move data from/in storage

Data systems

What are they, really?

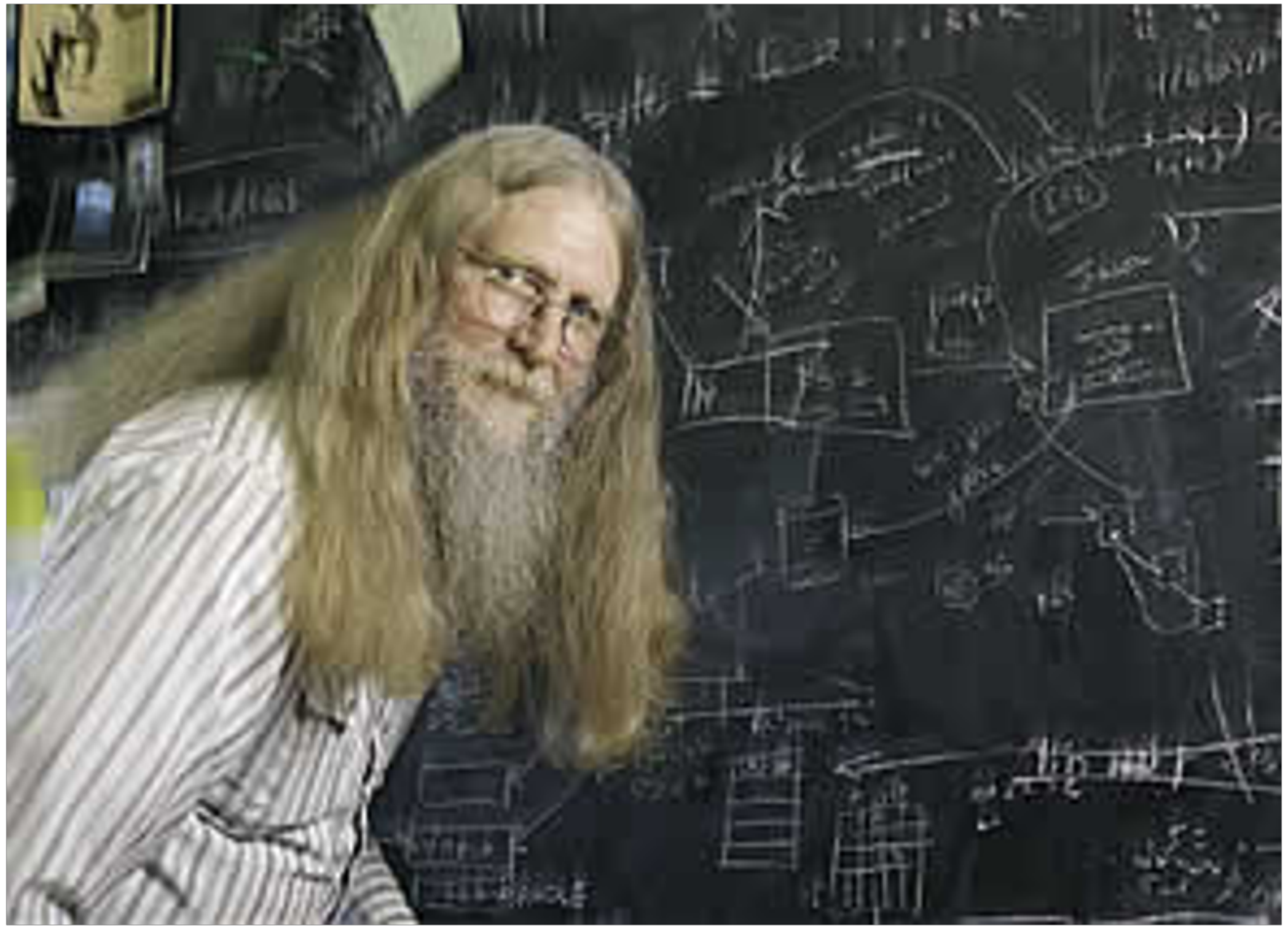


OK! But, don’t we have ?
Relational databases ?

Well, yes …

Relational databases

a.k.a relational data systems



Bruce Lyndsay, IBM Research

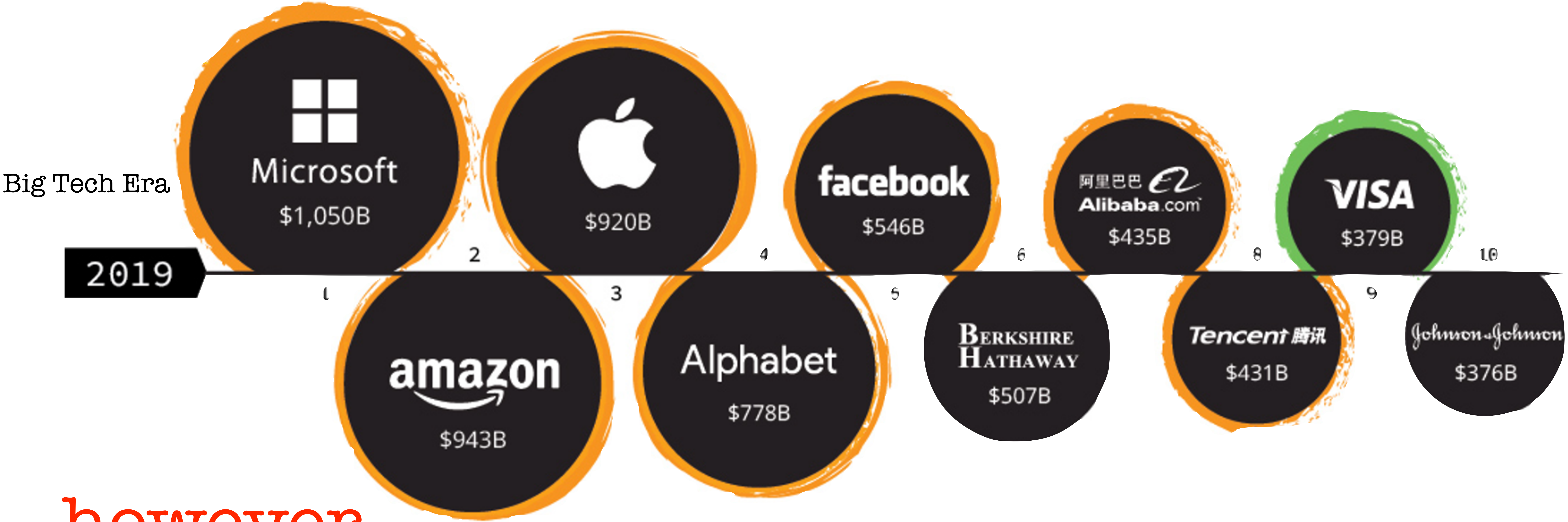
ACM SIGMOD Edgar F. Codd Innovations award 2012

> *Relational databases are the foundation of western civilization*

Brandeis UNIVERSITY

Relational databases

a.k.a relational data systems



... however

there's been a growing need for **tailored systems**

The need for **tailored systems**

One size DOES NOT fit all

growing
data size

new
hardware

heterogeneous
applications

new **performance**
goals

The need for **tailored systems**

One size DOES NOT fit all

growing
data size

new
hardware

heterogeneous
applications

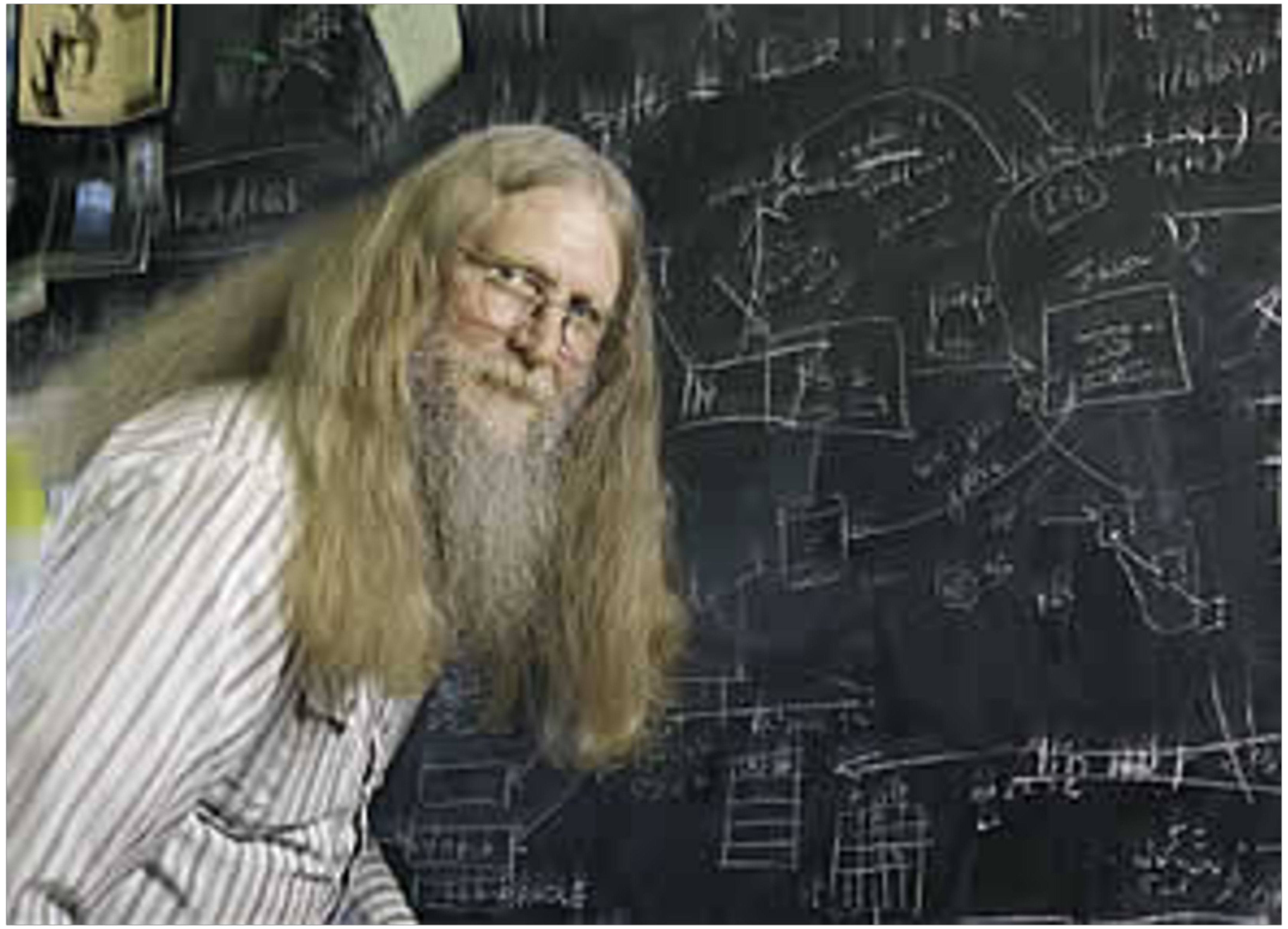
new **performance**
goals

Can **relational databases** not support
these requirements ?

Yes, but …

The need for **tailored systems**

One size DOES NOT fit all



Bruce Lyndsay, IBM Research

ACM SIGMOD Edgar F. Codd Innovations award 2012

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

Brandeis UNIVERSITY

The birth of **NoSQL**

A 2000's child

Not only SQL

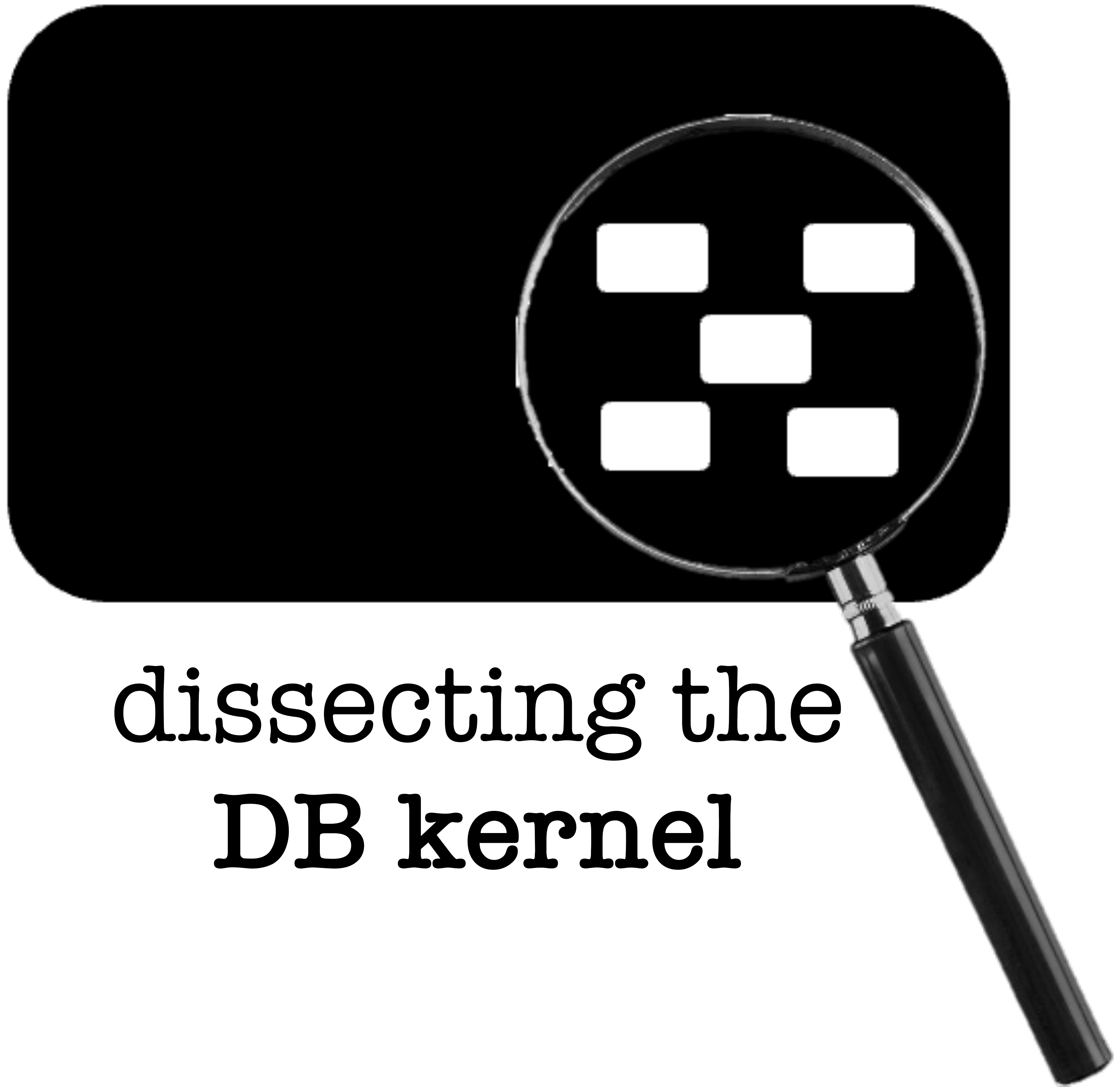
this is where we will spend our time!

steep competition to the **relational market**

since **early 2000's**

Understanding the **NoSQL kernel**

What's that all about?



dissecting the

DB kernel

system architecture (row/column/hybrid)

index design (tree, bitmap, trie, none)

hardware considerations (HDD, SSD, GPU)

performance optimization and **tuning**

optimizing **resource utilization**

Thought Experiment 3

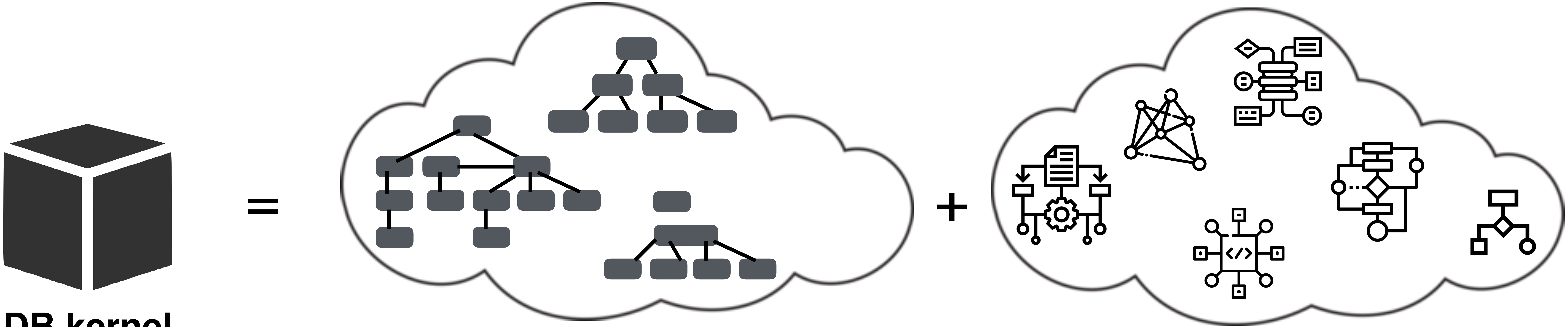
Example of **poor resource utilization**?

COSI 167A in G'Zang 121

Designing a DB kernel

A big undertaking!

designing a DB kernel is complex



DB kernel =

data structures

how to store data?

+

algorithms

how to access data?

Thought Experiment 4

So what makes designing kernels **complex**?

Designing a DB kernel

A big undertaking!

designing a DB kernel is **complex**

thousands of
data structures

x

hundreds of
tuning knobs

x

tens of thousands of
access methods

There are **more data structures** than there are **stars in the sky** at night!

Let’s **think** together

Identifying the design challenges

objective: Design a simple NoSQL kernel

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

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

workload has **reads** (*get*, *scan*, *range scan*) & **writes** (*put*) **interleaved**

How to **store** the data?

How to access **data** efficiently?

How to **delete** data?

Let's **think** together

Identifying the design challenges

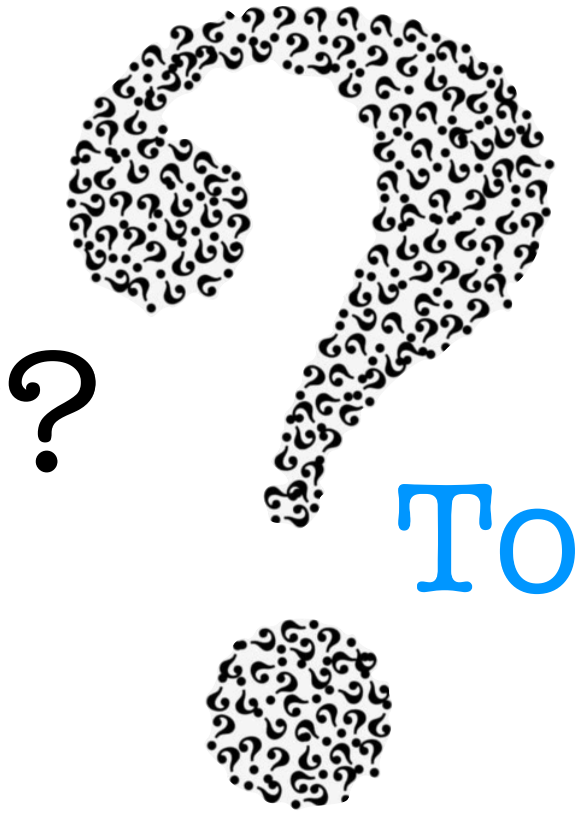
objective: Design a simple NoSQL kernel

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

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

workload has **reads** (*get*, *scan*, *range scan*) & **writes** (*put*) **interleaved**

How to **store** the data?
How to access **data** efficiently?
How to **delete** data?



Too many design choices!
Some conflicting!

Let’s **think** together

Identifying the design challenges

objective: Design a simple NoSQL kernel

design choices:

what is the **key/value**?

are they **stored together**?

can **read/write ratio change** over time?

what **index** to use? B-tree, hash-table, zonemaps, *none*?

how to handle (millions of) **concurrent queries**?

what happens if **data does not fit in memory**?

what about **privacy** and **security**?

how to offer **robustness** guarantees?

how to minimize **operational cost**?

Cost optimization

Operating on cloud

Operating on cloud brings a new set of challenges

large-scale deployment

millions of instances running in parallel

very different **performance tradeoffs**

Thought Experiment 5

10GB app: **1% less memory** in your machine. Do you care?

May be, may be not …

Cost optimization

Operating on cloud

Operating on cloud brings a new set of challenges

- **large-scale** deployment
- **millions of instances** running in parallel
- very different **performance tradeoffs**

Thought Experiment 6

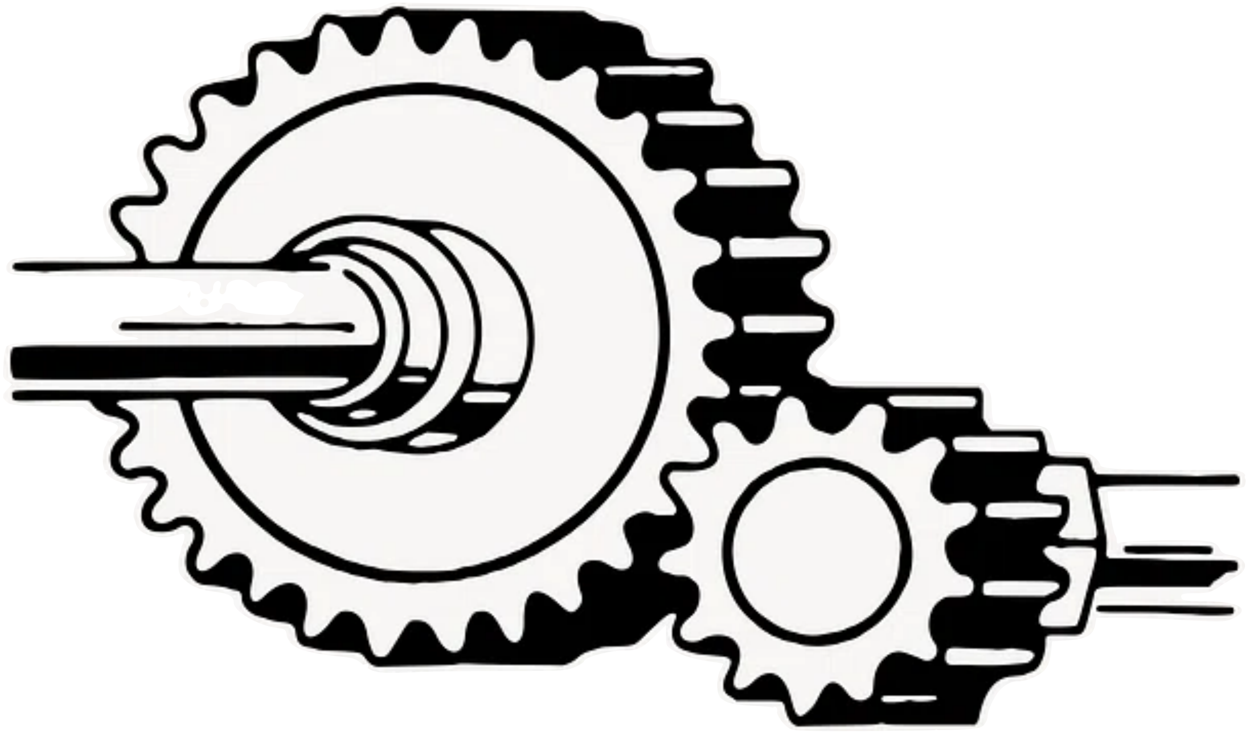
10GB app: **1% less memory** on **a million cloud instances**. Do you care?

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

~$800K in today's price

Goals of the class

Learning objectives



know the **internals of data systems**

understand system **design tradeoffs**

</>

sharpen your **systems skills**

data system designer & **researcher** are required

any **business**, any **startup**, any **scientific institution**

Summary

The key takeaways

NoSQL data stores are an **integral part** of today's data systems

key-value stores, **document** stores, **column** stores, **graph** stores

Key-value stores are the most commonly used NoSQL system

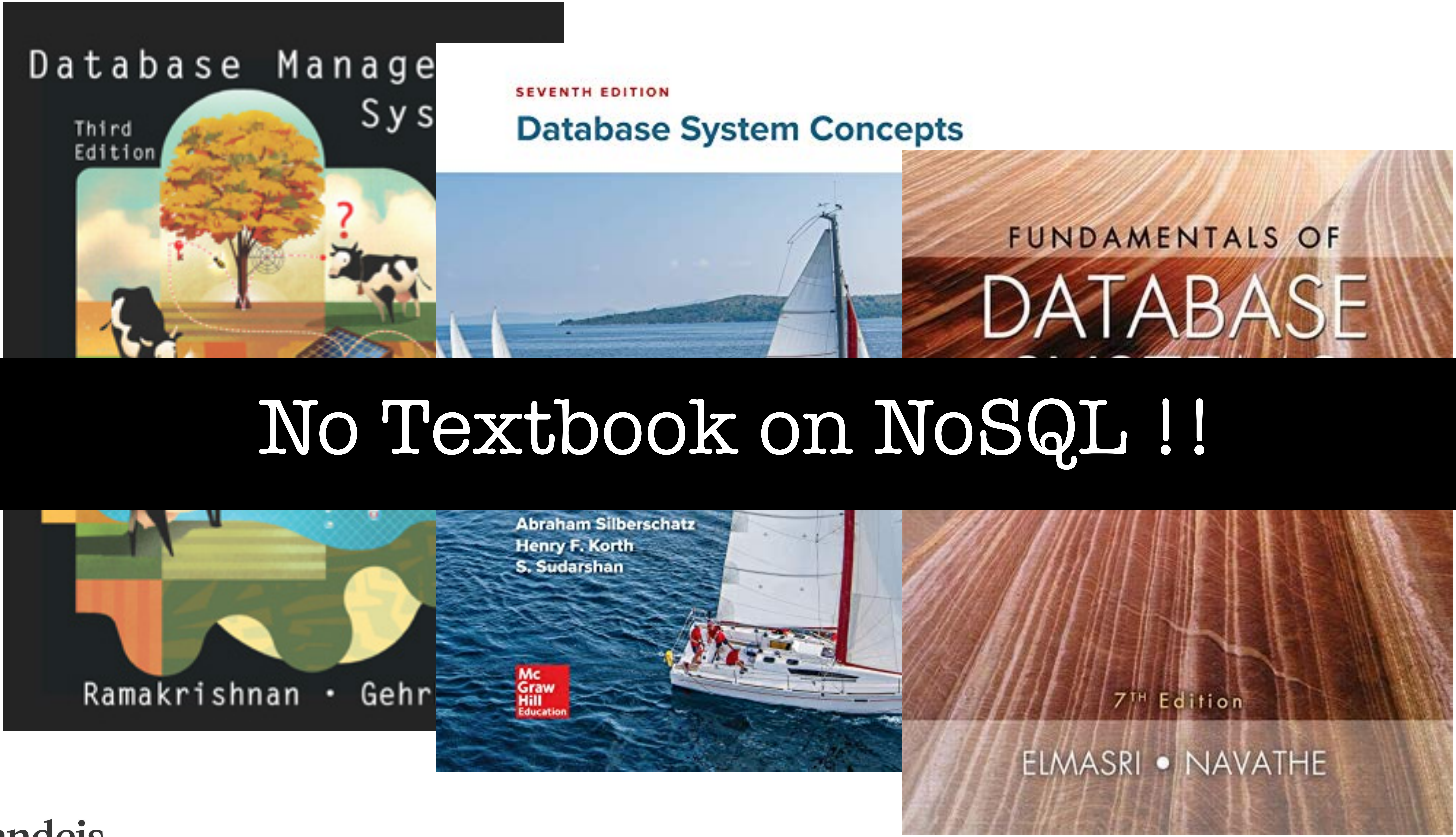
Simple yet **effective** data model; suitable for **heterogeneous data**

If you work with databases, **50-50** chance, you will work with **NoSQL**

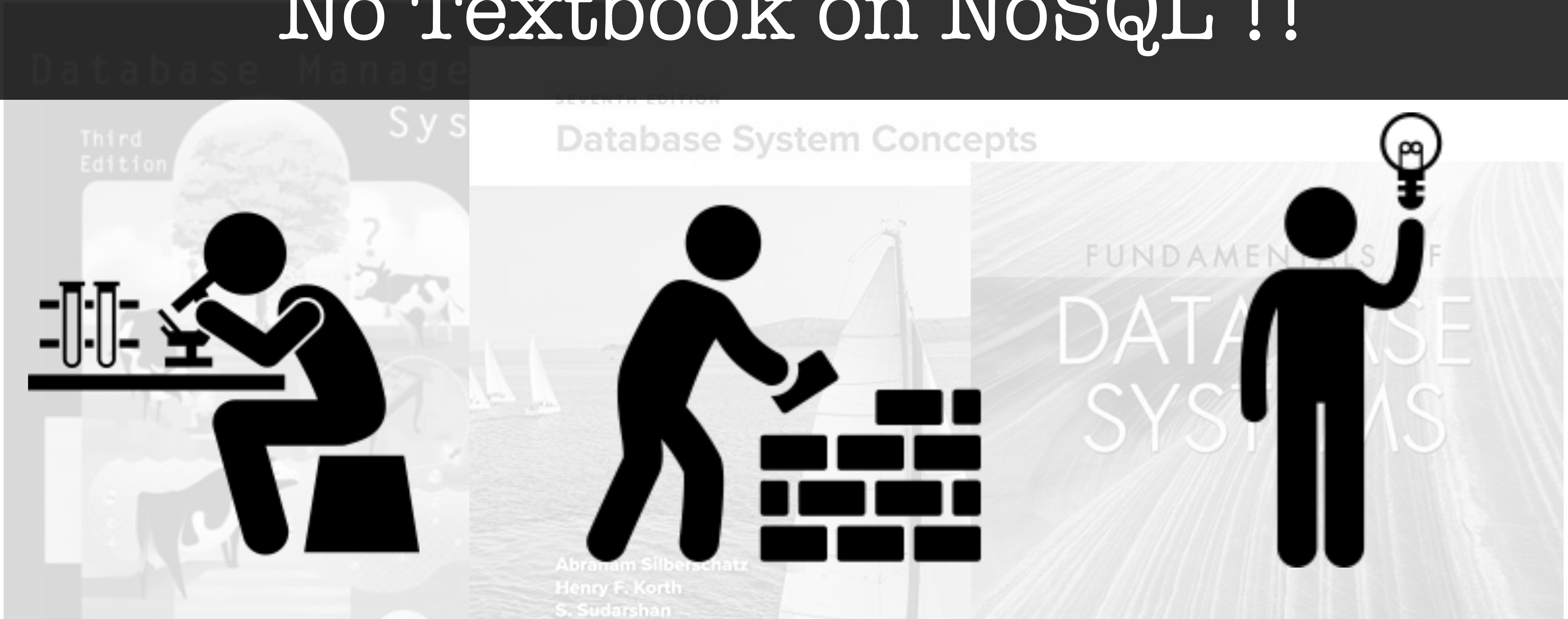
an **essential** skill to master

Unfortunately, …

No Textbook on NoSQL !!



No Textbook on NoSQL !!



explore NoSQL systems

understand the design space

hone your **systems skills**

We will read cutting-edge research papers!

Readings

Papers, papers, and papers

Architecture of a Database System
— J. Hellerstein, M. Stonebraker and J. Hamilton
Foundations and Trends in Databases, 2007

The Design and Implementation of Modern Column-store Database Systems
— D. Abadi, P. Boncz, S. Harizopoulos, S. Idreos, S. Madden
Foundations and Trends in Databases, 2013

Data Structures for Data-Intensive Applications
— Manos Athanassoulis, Stratos Idreos, and Dennis Shasha,
Foundations and Trends in Databases, 2023.

What to do **now**?

A) Go over the **syllabus** and the **class website**

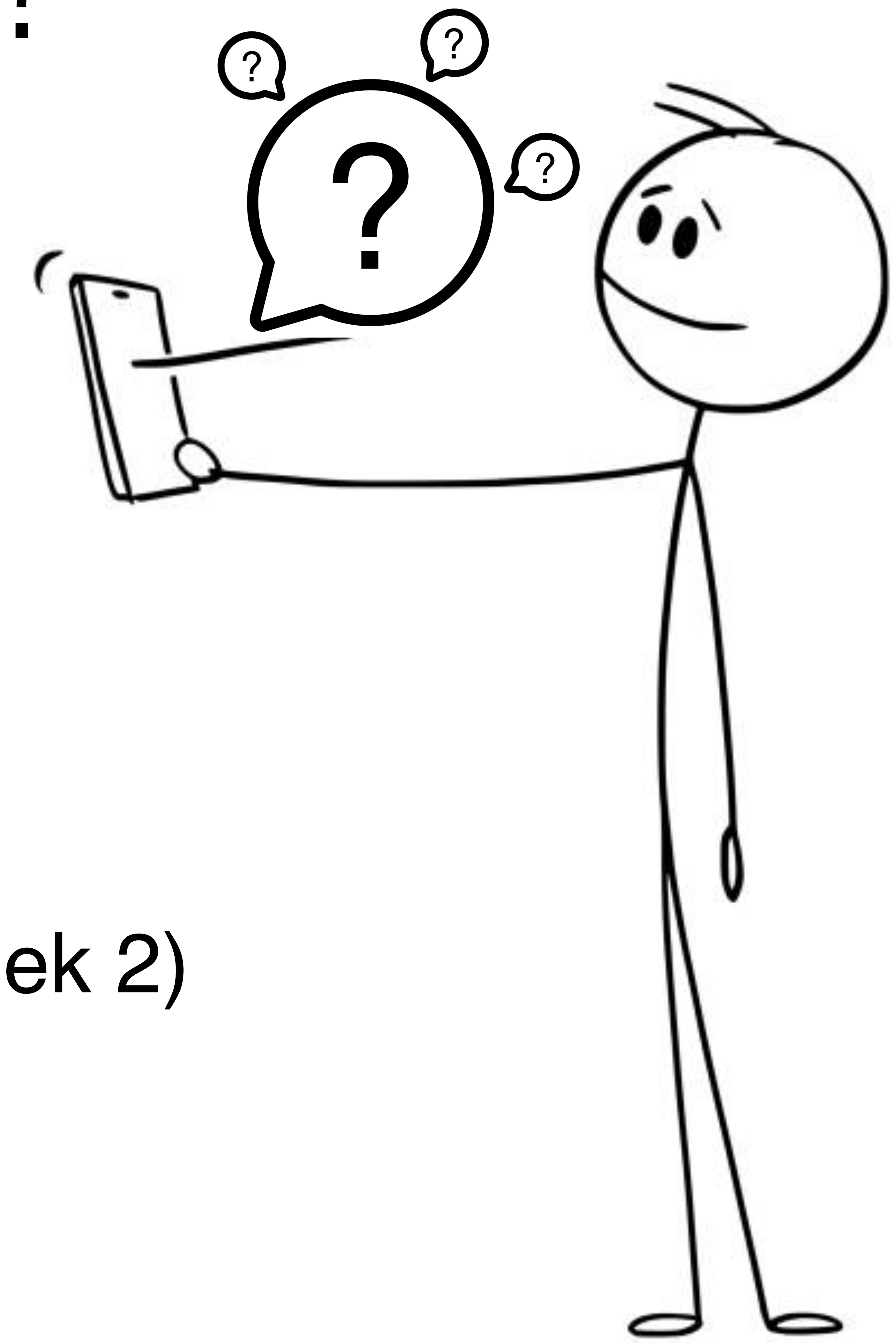
B) Register in **Gradescope** (code: **PYG88X**)

C) Be on the lookout for **Project 1** and start early!

D) Register for the **presentation** (week 2)

E) Submit **paper reviews** / **technical questions** (week 2)

E) **Recitations** are **optional**!



Next time in COSI 167A

Intro. + Administrivia

class **logistics**, **goals**, and **administrivia**

introduction to **NoSQL systems**

Project 1 details

COSI 127B
Database Management Systems

Class 1

Welcome to COSI 127B!

Prof. Subhadeep Sarkar

Brandeis
UNIVERSITY

https://ssd-brandeis.github.io/COSI-167A/