W4111
Introduction to Databases
Fall 2022

Computer Science Department

Columbia University

Welcome!

Eugene Wu

B.S.	U.C. Berkeley
Ph.D.	MIT
PostDoc	U.C. Berkeley
Professor	Columbia since Fall 2015

Database systems, vis, data analysis, cleaning, ML systems, crowdsourcing.

www.eugenewu.net
ewu@cs.columbia.edu
421 Mudd

Office hours
Thurs 12-1PM In-person/zoom.
or by appointment

Agenda

Overview

Course Info

Entity-Relationship Modeling

The Future of AI: How Artificial Intelligence Will Change the World

AI is constantly changing our world. Here are just a few ways AI will influence our lives.

Written by **Mike Thomas**

PEW RESEARCH CENTER | DECEMBER 10, 2018

Artificial Intelligence and the Future of Humans

Experts say the rise of artificial intelligence will make most people better off over the next decade, but many have concerns about how advances in AI will affect what it means to be human, to be productive and to exercise free will

BY JANNA ANDERSON AND LEE RAINIE

OPINION

Andrew Yang: Yes, Robots Are Stealing Your Job

Self-driving trucks will be great for the G.D.P. They'll be terrible for millions of truck drivers.

Nov. 14, 2019

EU Parliament
Draft Report on AI in the Digital Age

Argues that artificial intelligence (AI) is the key emerging technology within the fourth industrial revolution; notes that AI is the control centre of the new data layer that surrounds us and which can be thought of as the fifth element after air, earth, water and fire; states that by 2030, AI is expected to contribute more than EUR 11 billion to the global economy, an amount that almost matches China's GDP in 2020;

Conventional View of AI/Data Science

Lone data scientist uses a static, clean table,
applies statistics or fits an ML model
to increase a well-defined score

See popular ML articles, Kaggle competitions, etc

Conventional View of AI/Data Science

Lone data scientist uses a static, clean table,
applies statistics or fits an ML model
to increase a well-defined score

See popular ML articles, Kaggle competitions, etc

Conventional View of AI/Data Science

Team

Dynamic Messy Non-tabular

Lone data scientist uses a static, clean table,

applies statistics or fits an ML model

Unavailable

to increase a well-defined score

unclear, ill-defined

Huge amount of "unseen labor" (data engineering)

in order to support real-world data science & ML

In Reality…

An on-call engineer's biggest nightmare

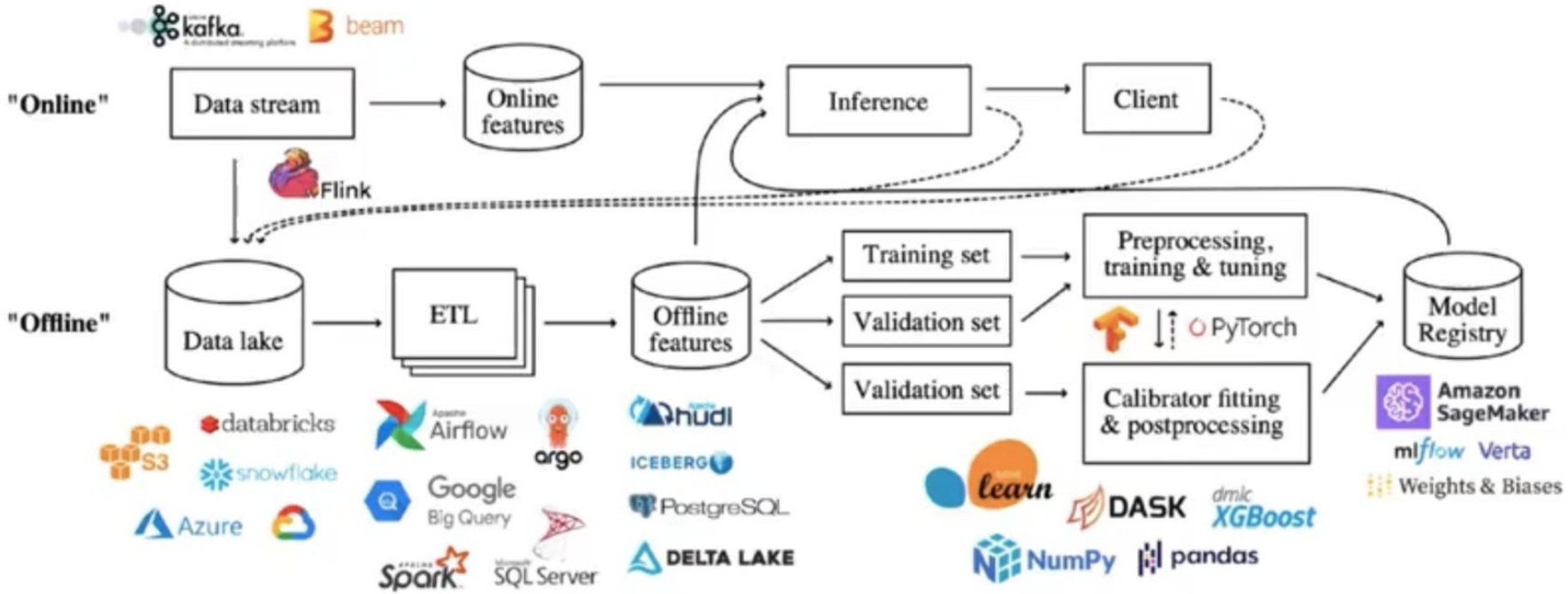


Figure 1: High-level architecture of a generic end-to-end machine learning pipeline. Logos represent a sample of tools used to construct components of the pipeline, illustrating heterogeneity in the tool stack. *Shankar et al. 2021*

https://www.facebook.com/Engineering/videos/1578607659138164/

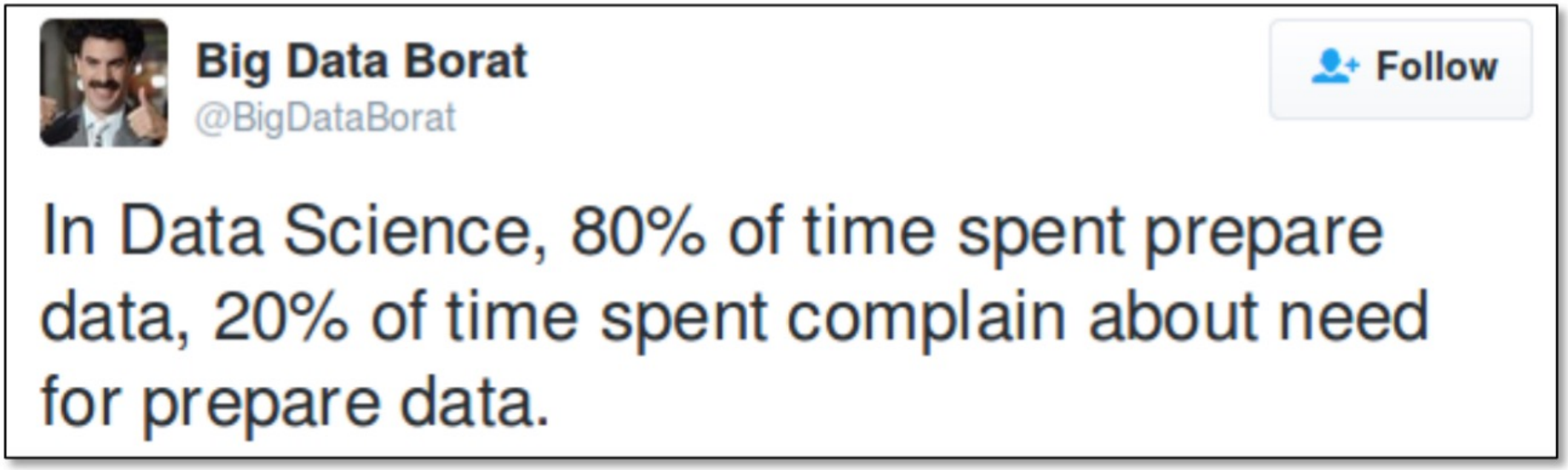
In Reality…

Data engineering dominates data science projects

Data engineer work >> data scientist work

Data engineering key to ML/AI/data science

Data Eng Dominates Data Science Projects



Most time spent on data engineering

Often not viewed as sexy, but critical

Data Eng Dominates Data Science Projects

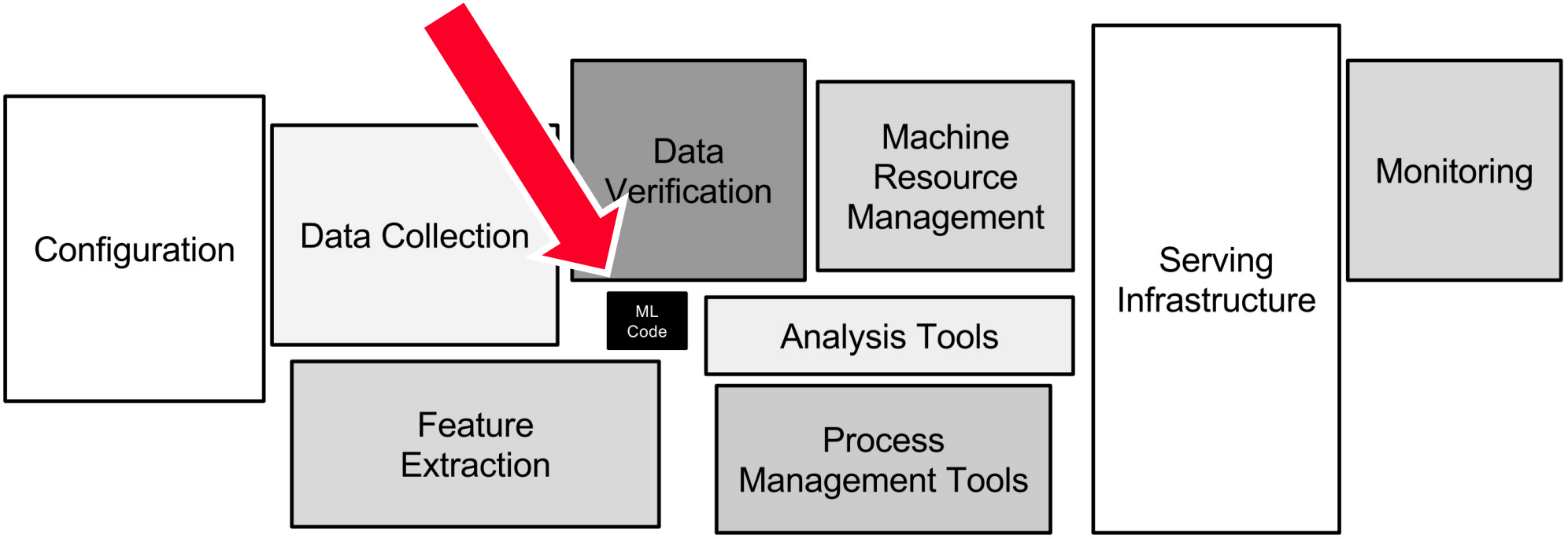
Hidden Technical Debt in Machine Learning Systems

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips
{dsculley,gholt,dgg,edavydov,toddphillips}@google.com
Google, Inc.

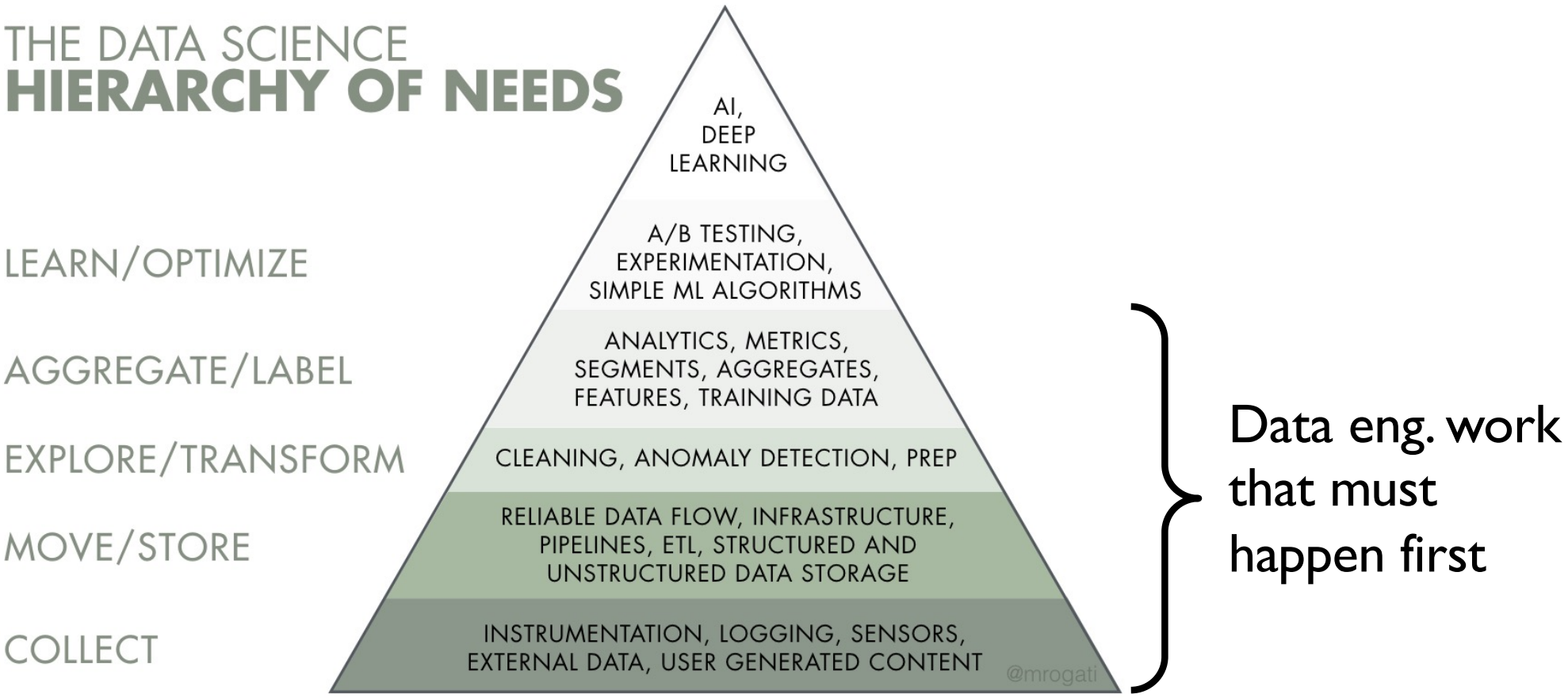
Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, Dan Dennison
{ebner,vchaudhary,mwyoung,jfcrespo,dennison}@google.com
Google, Inc.

Data Eng Dominates Data Science Projects

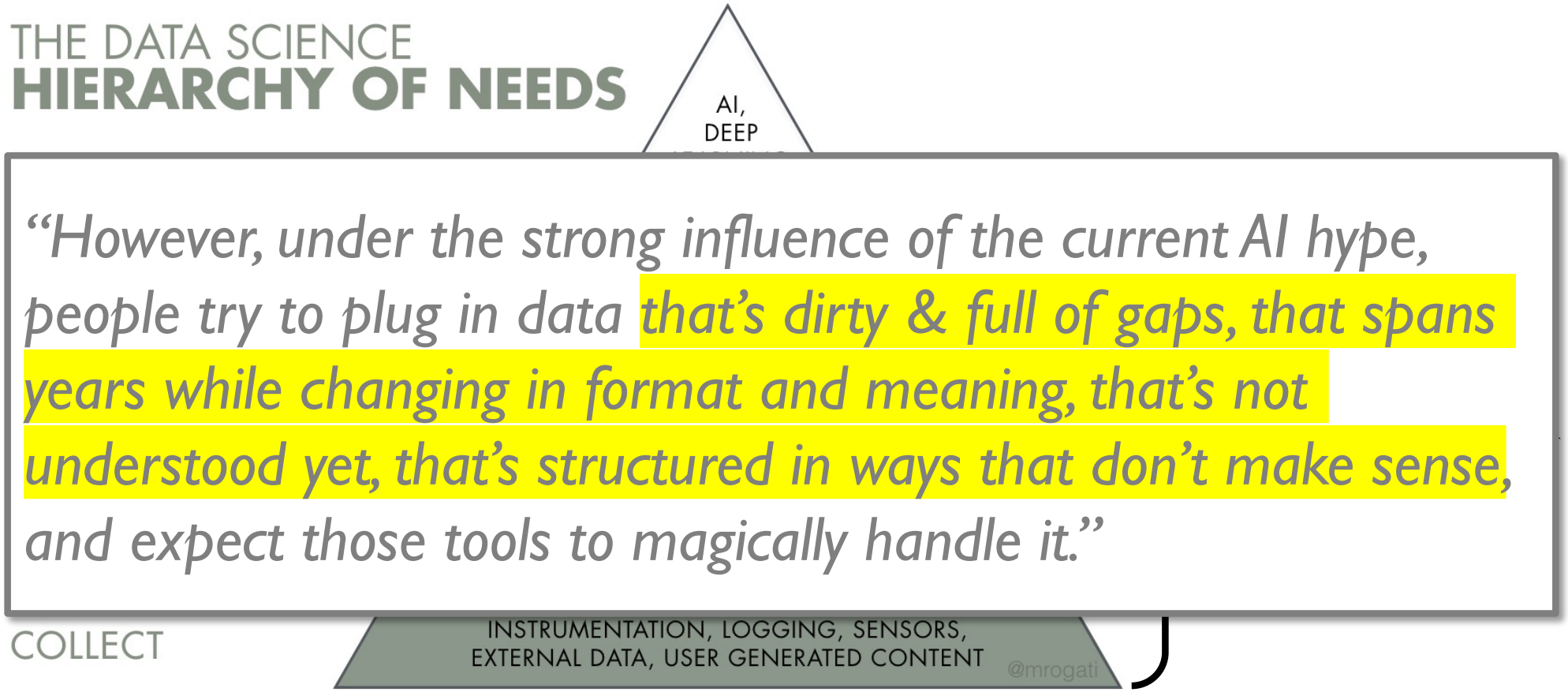
Hidden Technical Debt in Machine Learning Systems



Data Eng Dominates Data Science Projects



https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007



https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007

Data Engineering as a Job Category

"..70% more open roles at companies in data engineering as compared to data science…."

Data Engineering as a Job Category

Job Opening Estimates from Zippia

Data scientist:	79K	16% growth rate
Data engineer:	170K	21% growth rate

Overview of Data Engineering Concerns

ETL and Data Warehouses

Data lakes

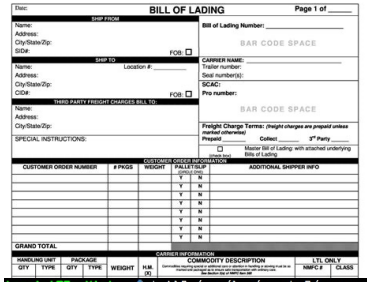
Data Quality

Metadata

Preparation

Python Scripts,
Spark/Databricks, etc

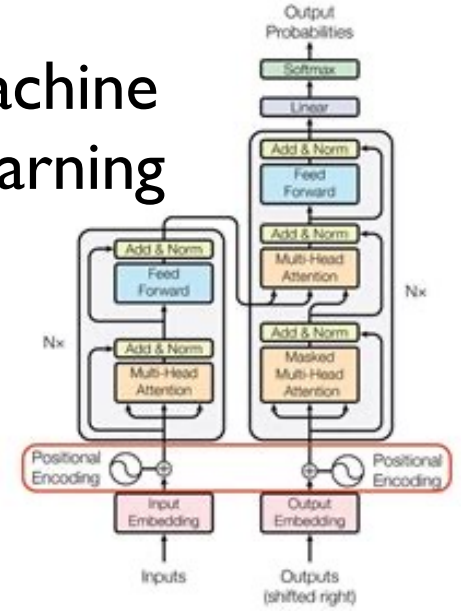
PDF



Data Preparation

Transform/clean data into useful form

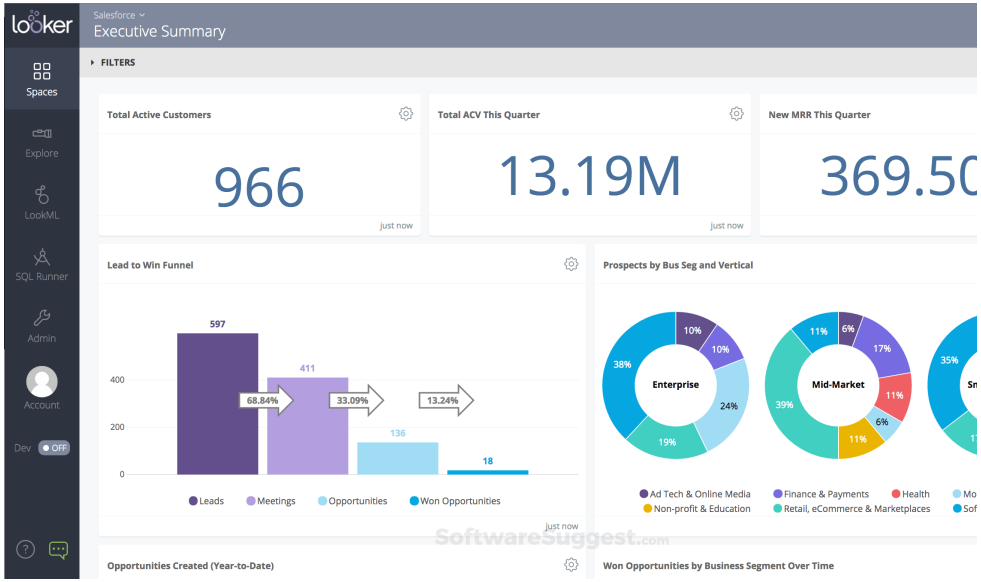
Machine Learning



Logs

```
tecmint@TecMint ~ $ tailf /var/log/apache2/access.log
127.0.0.1 - - [31/Oct/2017:11:11:37 +0530] "GET / HTTP/1.1" 200 729 "-" "Mozill
127.0.0.1 - - [31/Oct/2017:11:11:37 +0530] "GET /icons/blank.gif HTTP/1.1" 200
fox/56.0"
127.0.0.1 - - [31/Oct/2017:11:11:37 +0530] "GET /icons/folder.gif HTTP/1.1" 200
efox/56.0"
127.0.0.1 - - [31/Oct/2017:11:11:37 +0530] "GET /icons/text.gif HTTP/1.1" 200 5
ox/56.0"
127.0.0.1 - - [31/Oct/2017:11:11:38 +0530] "GET /favicon.ico HTTP/1.1" 404 500
127.0.0.1 - - [31/Oct/2017:11:12:05 +0530] "GET /tecmint/ HTTP/1.1" 200 787 "ht
0"
127.0.0.1 - - [31/Oct/2017:11:12:05 +0530] "GET /icons/back.gif HTTP/1.1" 200 4
01 Firefox/56.0"
127.0.0.1 - - [31/Oct/2017:11:13:58 +0530] "GET /tecmint/Videos/ HTTP/1.1" 200
101 Firefox/56.0"
127.0.0.1 - - [31/Oct/2017:11:13:58 +0530] "GET /icons/compressed.gif HTTP/1.1"
) Gecko/20100101 Firefox/56.0"
127.0.0.1 - - [31/Oct/2017:11:13:58 +0530] "GET /icons/movie.gif HTTP/1.1" 200
o/20100
```

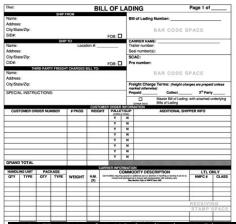
Visualization

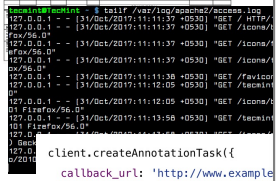


JSON

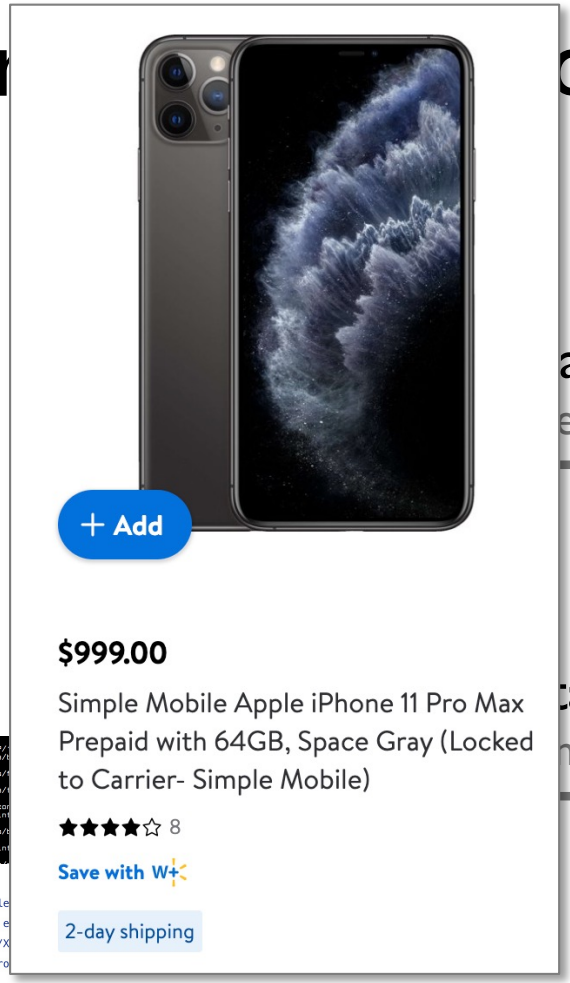
```
client.createAnnotationTask({
  callback_url: 'http://www.example.com/callback',
  instruction: 'Draw a box around each rooftop and pool.',
  attachment: 'http://i.imgur.com/XOJbalC.jpg',
  objects_to_annotate: ['pool', 'rooftop'],
  with_labels: true,
  min_width: 30,
  min_height: 30
}, (err, task) => {
    // do something with task
});
```

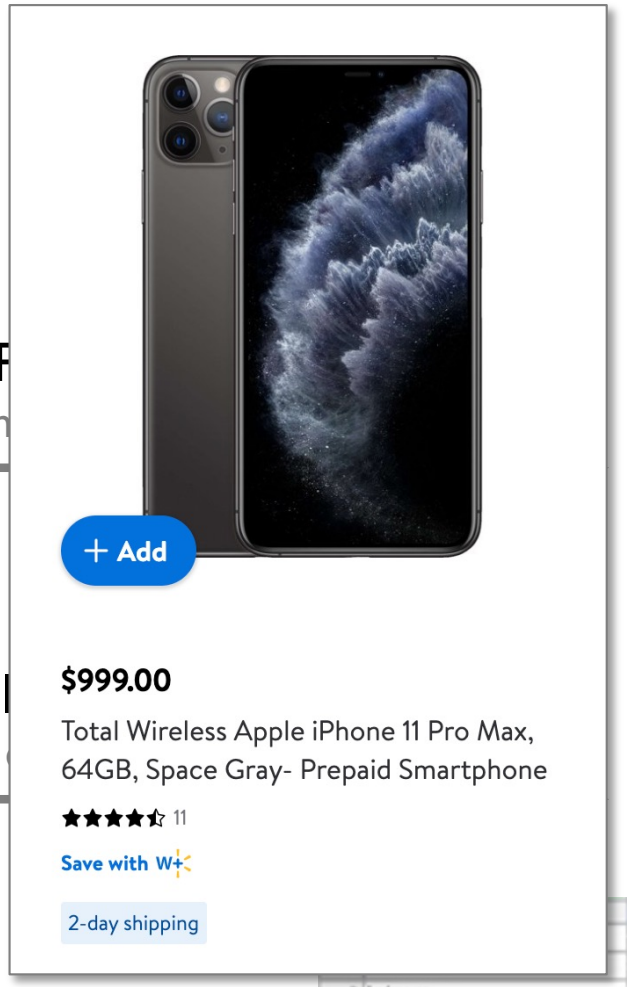
Sources





```
client.createAnnotationTask({
  callback_url: 'http://www.example
  instruction: 'Draw a box around e
  attachment: 'http://i.imgur.com/X
  objects_to_annotate: ['pool', 'ro
  with_labels: true,
  min_width: 30,
  min_height: 30
}, (err, task) => {
  // do something with task
}
```





Use Cases

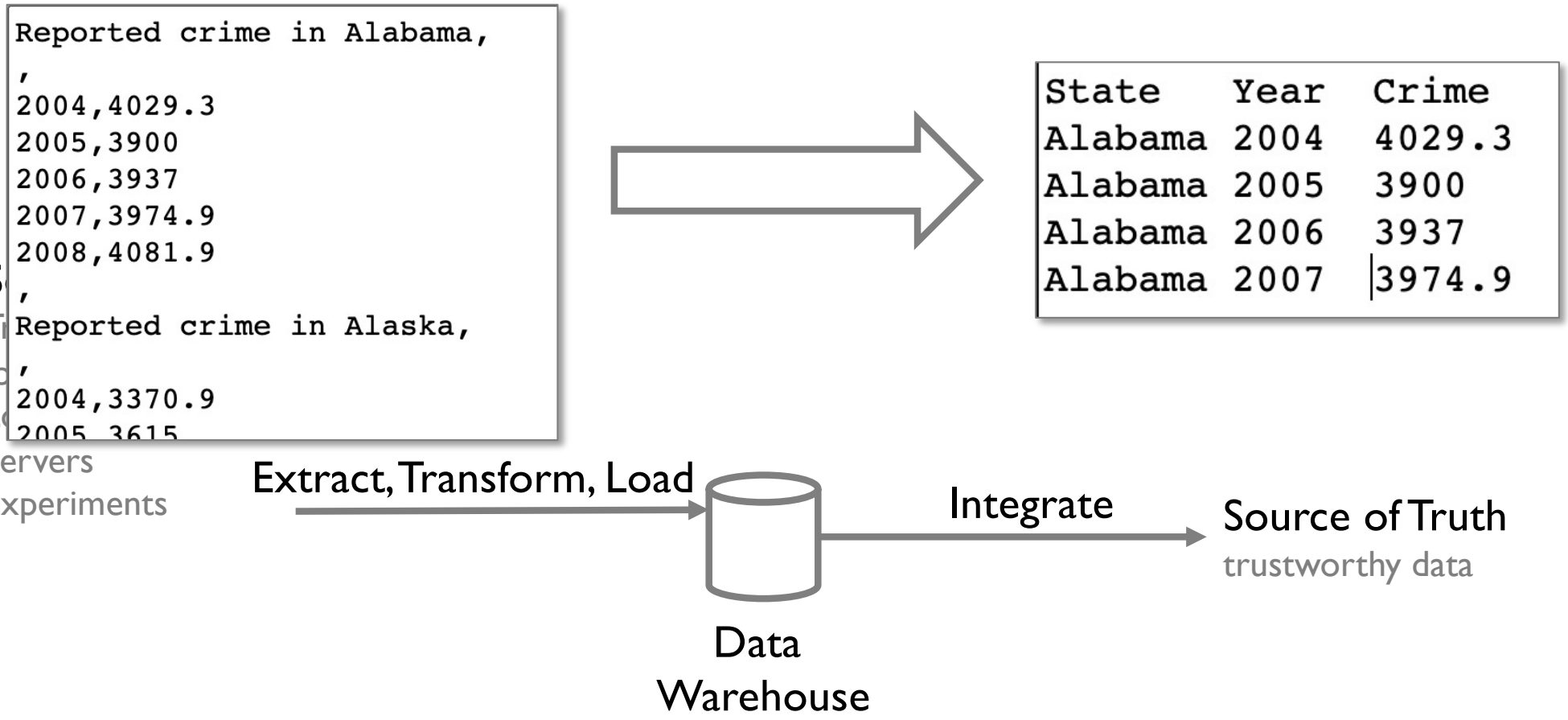
AI, ML, Data science, Apps, Webservices

Source of Truth

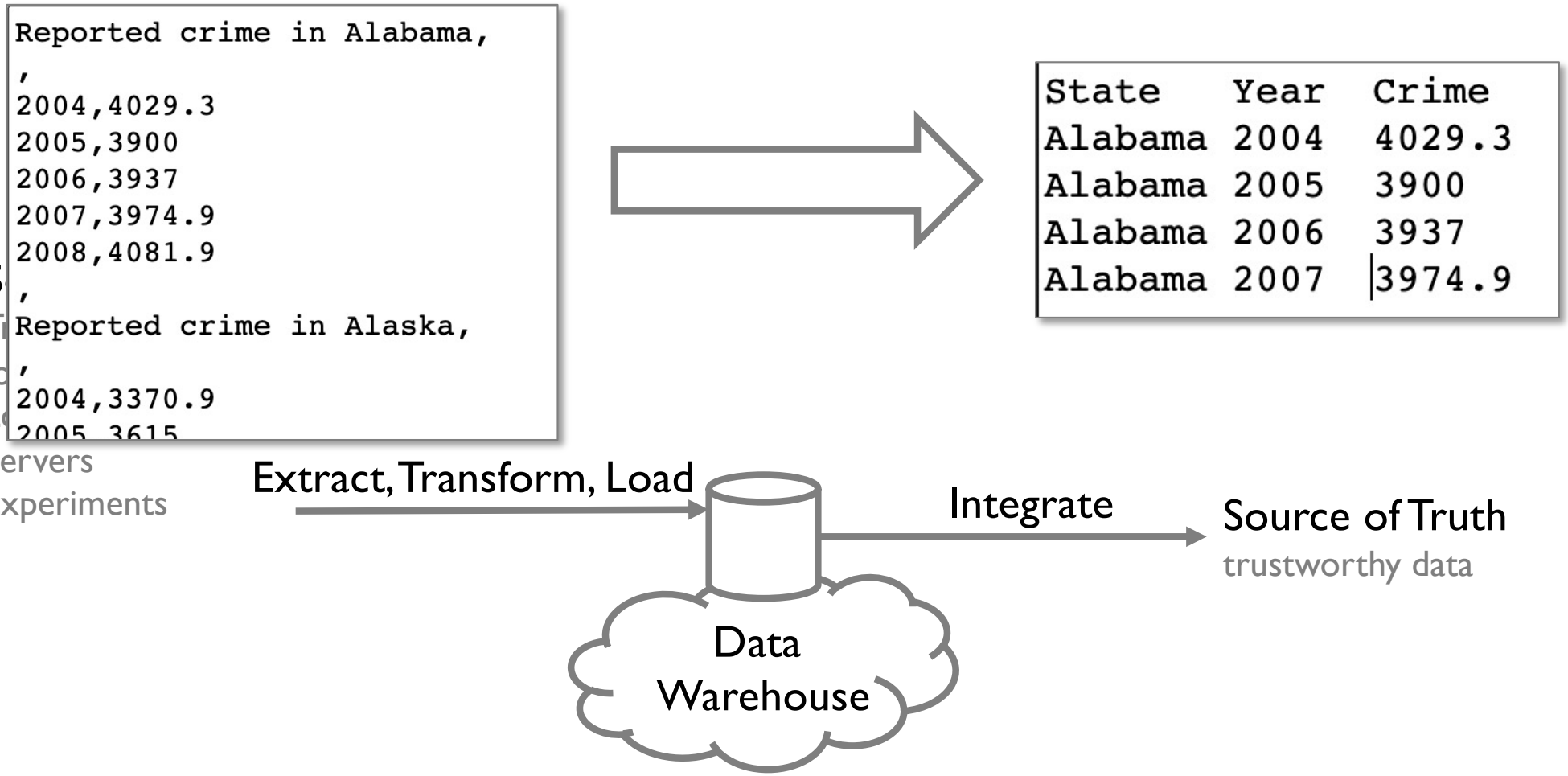
trustworthy data

		2004	2005
		4029.3	3900
		3370.9	3615
2	Arizona	5073.3	4827
3	Arkansas	4033.1	4068
4	California	3423.9	3321
5	Colorado	3918.5	4041
6	Connecticut	2684.9	2579
7	Delaware	3283.6	3110
8	District of Columbia	4852.8	4490
9	Florida	4182.5	4013

ETL: Extract Transform Load (1980s)



ELT: Extract Load Transform (2010s)



Servers
Experiments

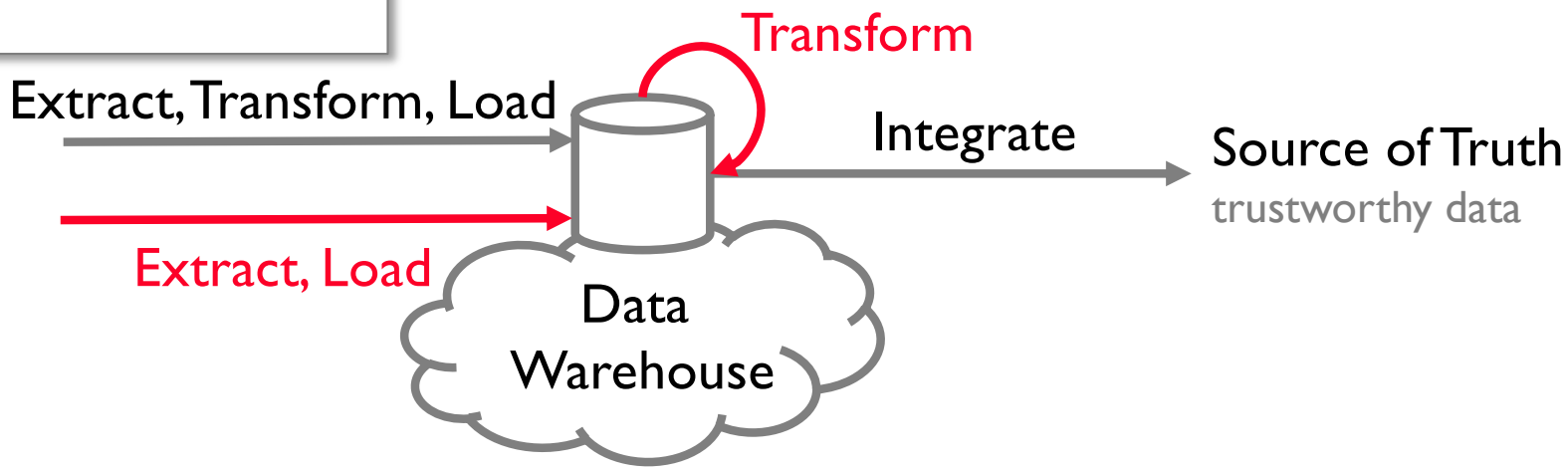
ELT: Extract Load Transform (2010s)

```
Reported crime in Alabama,
,
2004,4029.3
2005,3900
2006,3937
2007,3974.9
2008,4081.9
,
Reported crime in Alaska,
,
2004,3370.9
2005,3615
```

SQL Transform

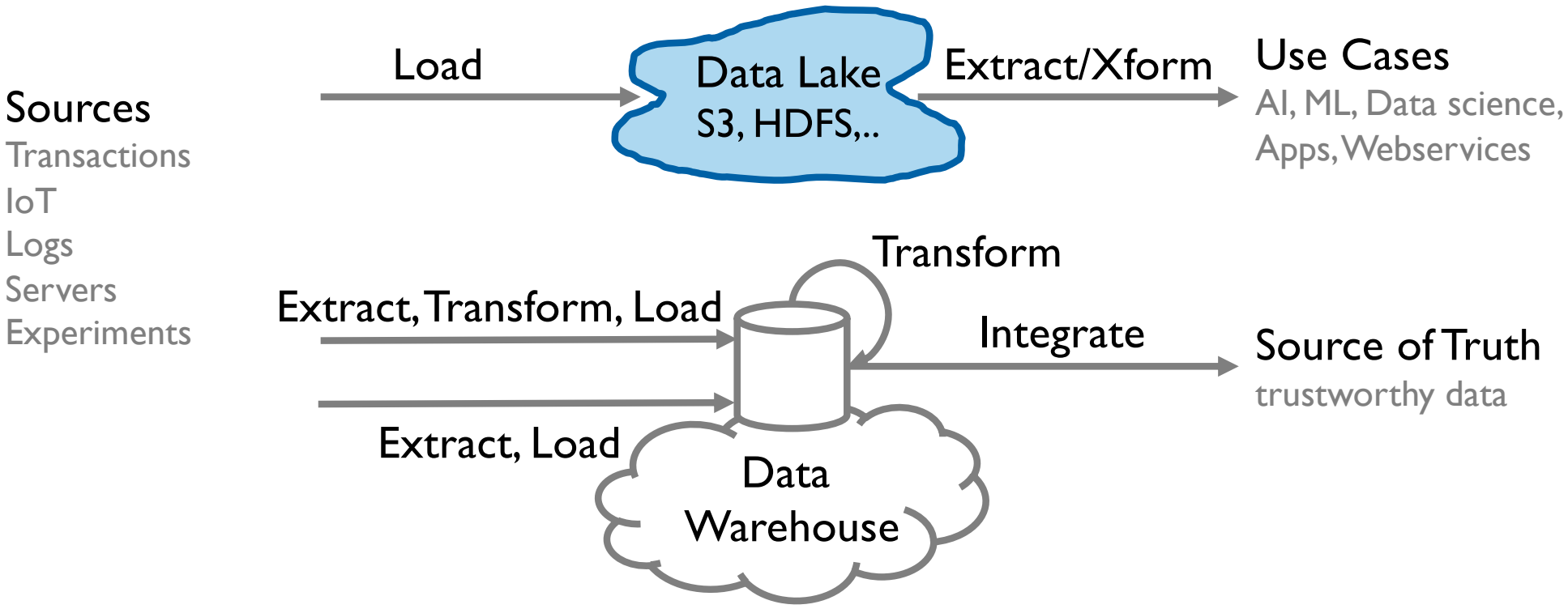
State	Year	Crime
Alabama	2004	4029.3
Alabama	2005	3900
Alabama	2006	3937
Alabama	2007	3974.9

Servers
Experiments

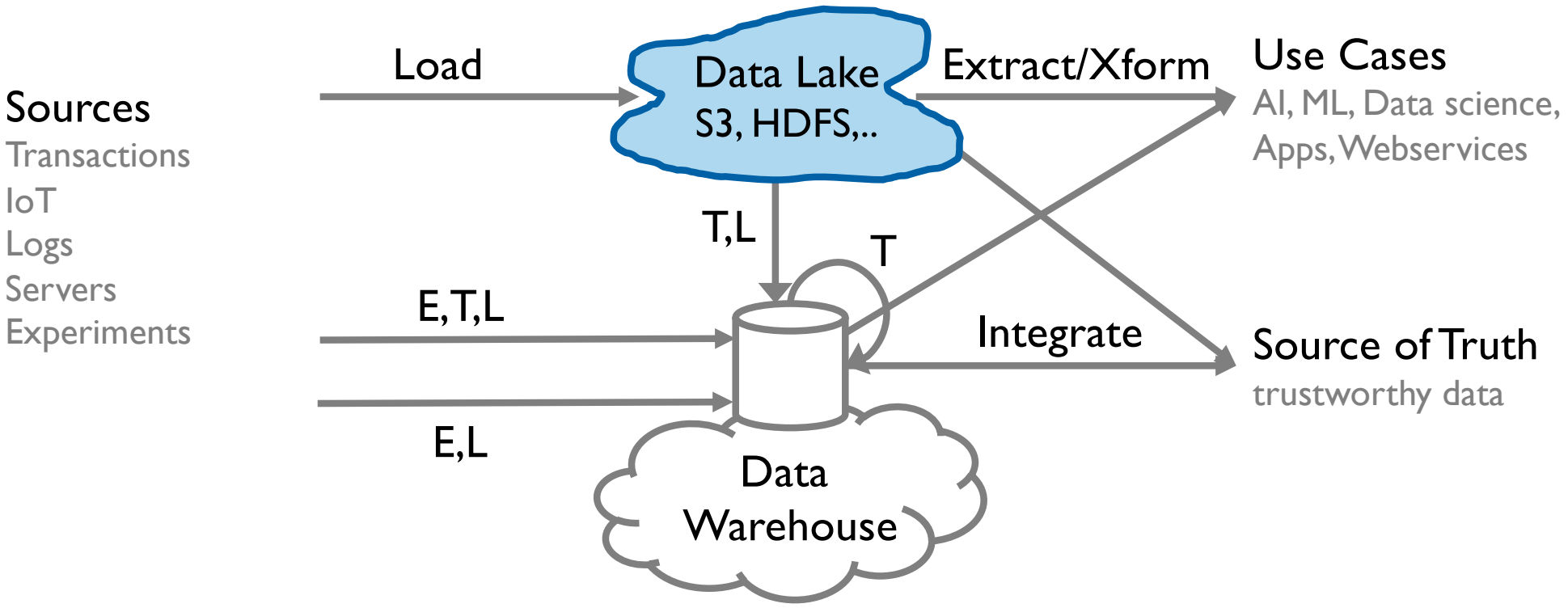


Data Lakes (2000s)

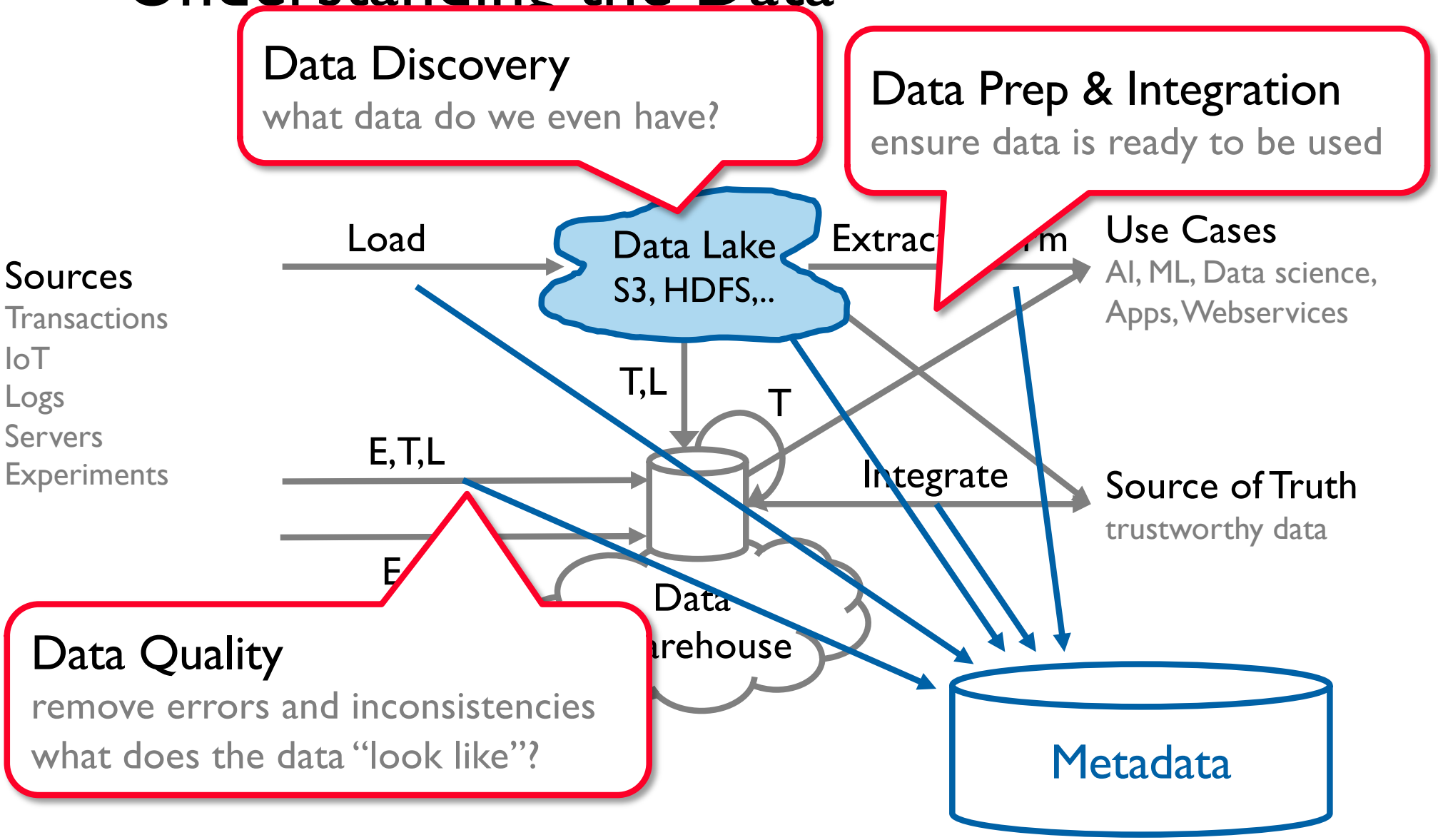
Store as files, transform when needed

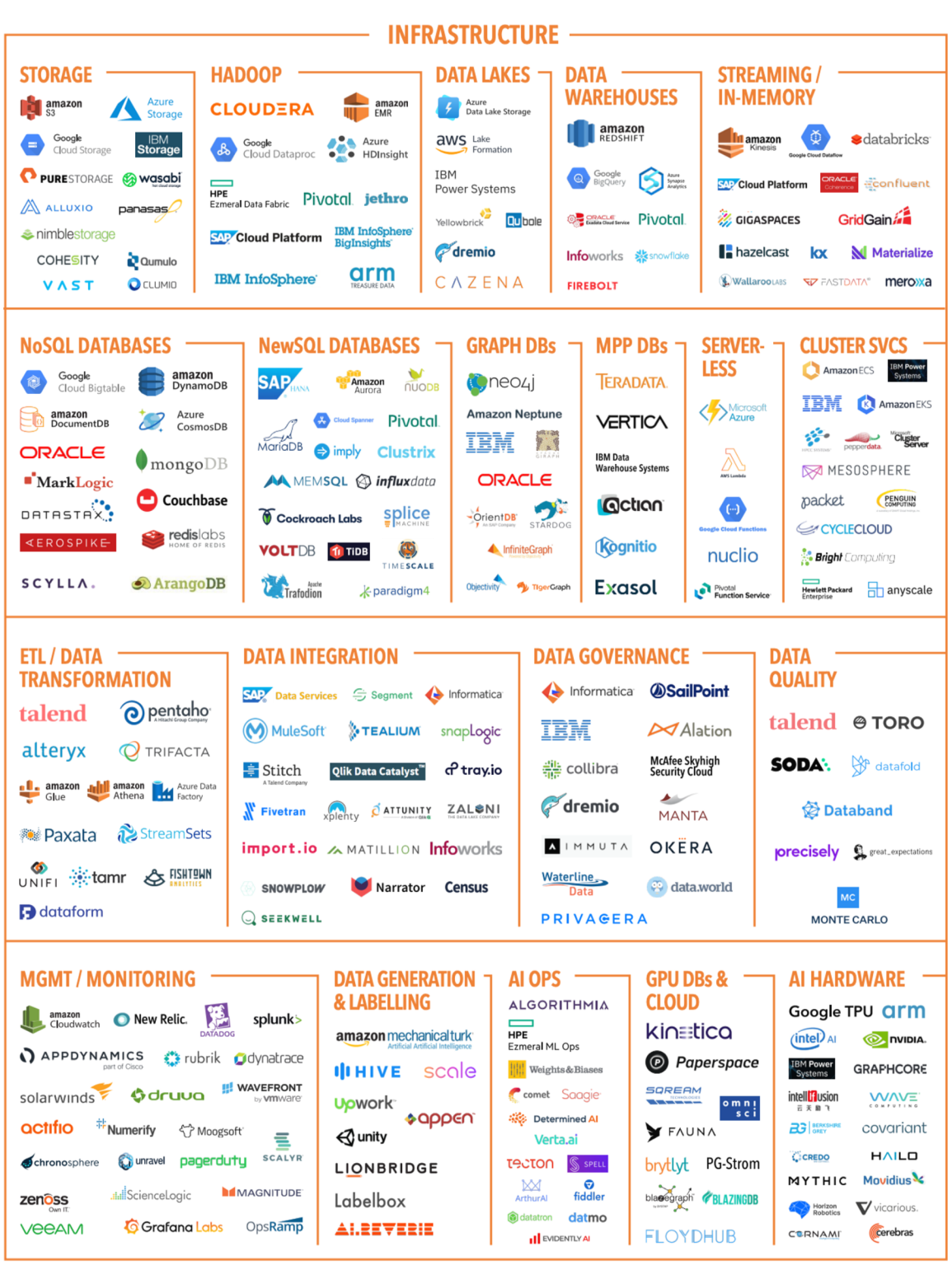


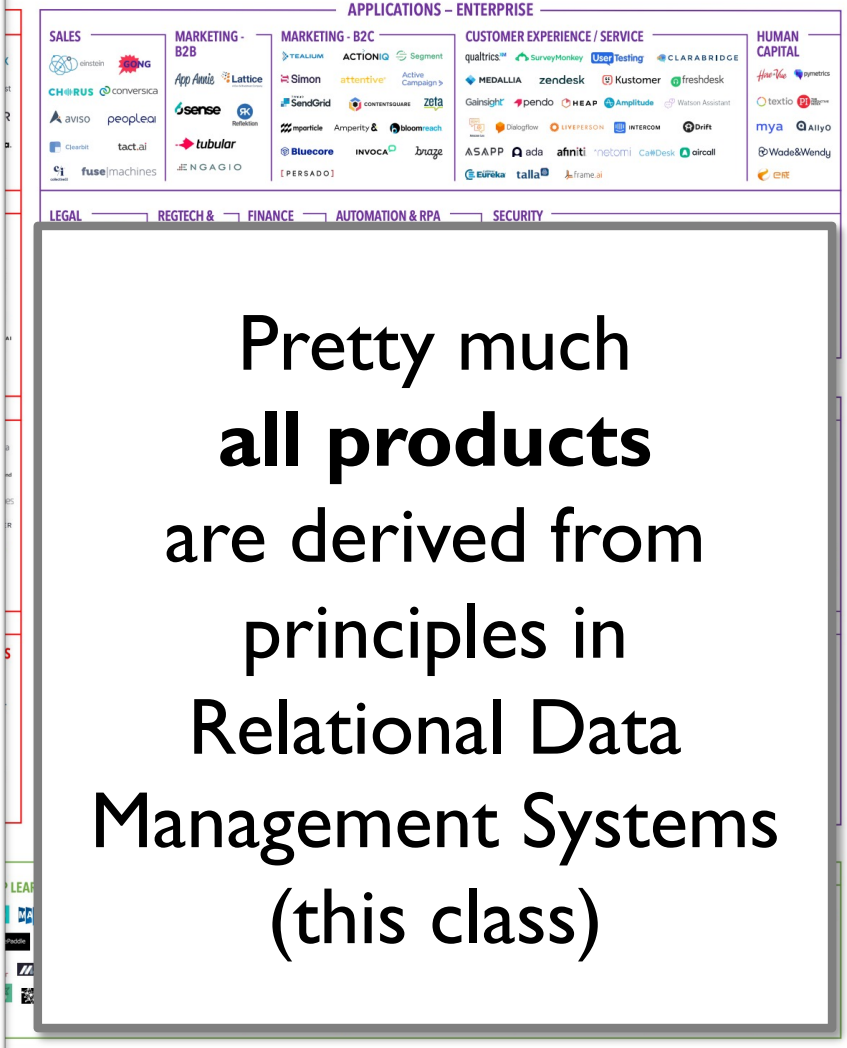
Everything Everywhere All At Once (2010s)



Understanding the Data







Pretty much
all products
are derived from
principles in
Relational Data
Management Systems
(this class)

tturck.com/data2020

FIRSTMARK
EARLY STAGE VENTURE CAPITAL

https://mattturck.com/data2020/

4111: Intro to Relational Data Management Systems

What's a database?

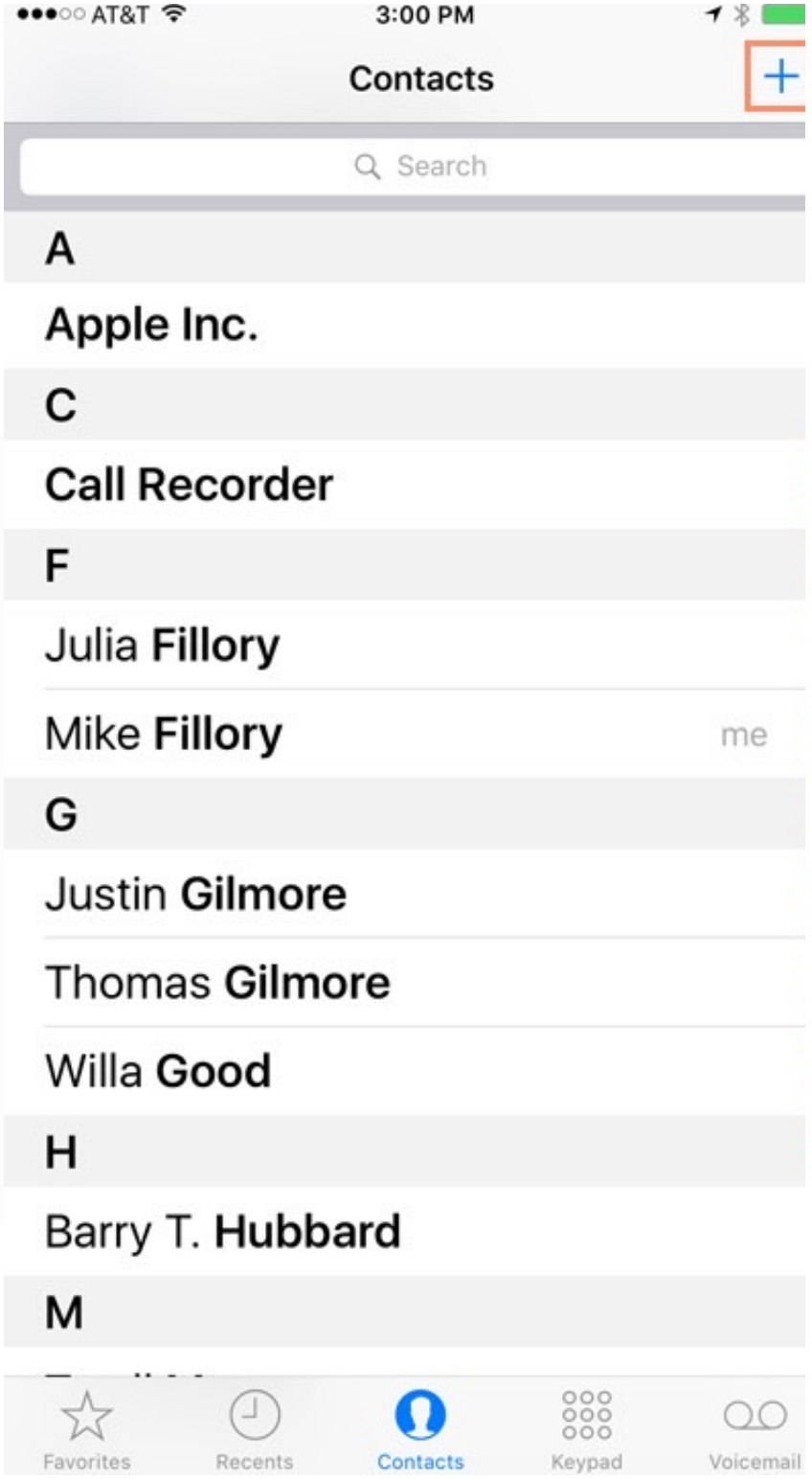
What's a database management system (DBMS)?

What are the core ideas?

What is a Database?

	A	B	C	D	E	F	G	H	I	J	K
1	color	date	slug	title	lshow	link	readings	optional	assigned	ashow	due
2	white	21-Jan	Intro + ER Models			https://githu	Ch 1, 2		HW 0		
3	#e7f8ff	28-Jan	ER Models				Ch 2		<a href="https://github.com/w4111/hw1-s22">H	0	HW 0
4	#e7f8ff	4-Feb	Data Models				Ch 3	optional: <a		0	HW1 Part1
5	#e7f8ff	11-Feb	Data Models + ER->Relational				Ch 3	optional: <a		0	Project 1 Part 1 approval phase
6	#f2f9ed	18-Feb	Relational Algebra				Ch 4		<a href="https://github.com/w4111/project1">Pr	0	Project 1 Part 1 approval phase
7	#f2f9ed	25-Feb	SQL: Basics				Ch 5			0	HW1 Part 2
8	#f2f9ed	4-Mar	SQL: Advanced				Ch 5			0	HW2
9	white	11-Mar	Midterm	one 8x11 page cheat sheet both sides					<a href="https://github.com/w4111/project1/blol	0	
10	white	18-Mar	HOLIDAY							0	Project 1 Part 2
11	#edf3f9	25-Mar	APIs				Ch 6			0	
12	#edf3f9	1-Apr	Data Quality	Normalization and data errors			Ch 19		<a href="https://github.com/w4111/hw4-s22">H	0	HW3
13	#ddf9ff	8-Apr	Physical Design				Ch 8		<a href="https://github.com/w4111/project2_s22	0	
14	#ddf9ff	15-Apr	Query Processing				Ch 12			0	Project 1 Part 3
15	#ddf9ff	22-Apr	Transactions				Ch 16, 18			0	
16	white	29-Apr	Data Pipelines							0	HW 4
17	white	13-May	Exam 2 (Cum	one 8x11 page cheat sheet both sides						0	Project 2
18											
19											
20											

What is a Database?



What is a Database?

```
2012-01-04 00:01:23,180 INFO org.apache.hadoop.hdfs.server.datanode.DataNode: Receiving block blk_-2281137920769
010
2012-01-04 00:01:23,184 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /127.0.0.1:32981,
 cliID: DFSClient_-603743753, offset: 0, srvID: DS-292194659-127.0.1.1-50010-1324763300176, blockid: blk_-228113
2012-01-04 00:01:23,185 INFO org.apache.hadoop.hdfs.server.datanode.DataNode: PacketResponder 0 for block blk_-2
2012-01-04 00:01:23,291 INFO org.apache.hadoop.hdfs.server.datanode.DataNode: Receiving block blk_37660314352523
10
2012-01-04 00:01:23,293 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /127.0.0.1:32982,
cliID: DFSClient_-603743753, offset: 0, srvID: DS-292194659-127.0.1.1-50010-1324763300176, blockid: blk_37660314
2012-01-04 00:01:23,293 INFO org.apache.hadoop.hdfs.server.datanode.DataNode: PacketResponder 0 for block blk_37
2012-01-04 00:01:23,324 INFO org.apache.hadoop.hdfs.server.datanode.DataNode: Receiving block blk_-8044922265890
010
2012-01-04 00:01:23,326 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /127.0.0.1:32983,
liID: DFSClient_-603743753, offset: 0, srvID: DS-292194659-127.0.1.1-50010-1324763300176, blockid: blk_-80449222
2012-01-04 00:01:23,327 INFO org.apache.hadoop.hdfs.server.datanode.DataNode: PacketResponder 0 for block blk_-8
2012-01-04 00:01:23,409 INFO org.apache.hadoop.hdfs.server.datanode.DataNode: Receiving block blk_-9657937572621
10
2012-01-04 00:01:23,411 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /127.0.0.1:32984,
, cliID: DFSClient_-603743753, offset: 0, srvID: DS-292194659-127.0.1.1-50010-1324763300176, blockid: blk_-96579
2012-01-04 00:01:23,411 INFO org.apache.hadoop.hdfs.server.datanode.DataNode: PacketResponder 0 for block blk_-9
2012-01-04 00:01:23,433 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /127.0.0.1:50010,
 cliID: DFSClient_-2054881890, offset: 0, srvID: DS-292194659-127.0.1.1-50010-1324763300176, blockid: blk_-96579
2012-01-04 00:01:23,494 INFO org.apache.hadoop.hdfs.server.datanode.DataNode: Receiving block blk_54159109576590
10
2012-01-04 00:01:23,498 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /127.0.0.1:32987,
, cliID: DFSClient_-2054881890, offset: 0, srvID: DS-292194659-127.0.1.1-50010-1324763300176, blockid: blk_54159
2012-01-04 00:01:23,498 INFO org.apache.hadoop.hdfs.server.datanode.DataNode: PacketResponder 0 for block blk_54
2012-01-04 00:01:23,523 INFO org.apache.hadoop.hdfs.server.datanode.DataNode: Receiving block blk_-5517241460358
```

What is a Database?

What is a Database?

Lots of

Structured data

Database Management System (DBMS)

A system to store, manage and access databases

Database Management System (DBMS)

System to safely and reliably store lots of persistent structured data and is convenient for multiple users to efficiently access and modify.

Is a script a DBMS?

Javascript/Python Script

- Data stored in variables (RAM)
- Very fast access
- Data structures (lists, dicts, tuples)

Is Excel a DBMS?

Microsoft office security

Visually access/modify/compute over data cells

Click save to store persistently

Is the file system a DBMS?

Manages files that are persistently stored on disk

Open/read/seek/write access to files

Access via file names

Access control via permissions

Is the file system a DBMS?

You and a friend edit the same text file

Save at the same time

What happens?

1. Your changes survive
2. Friend's changes survive
3. Both changes survive
4. No changes survive
5. ¯_(ツ)_/¯

Is the file system a DBMS?

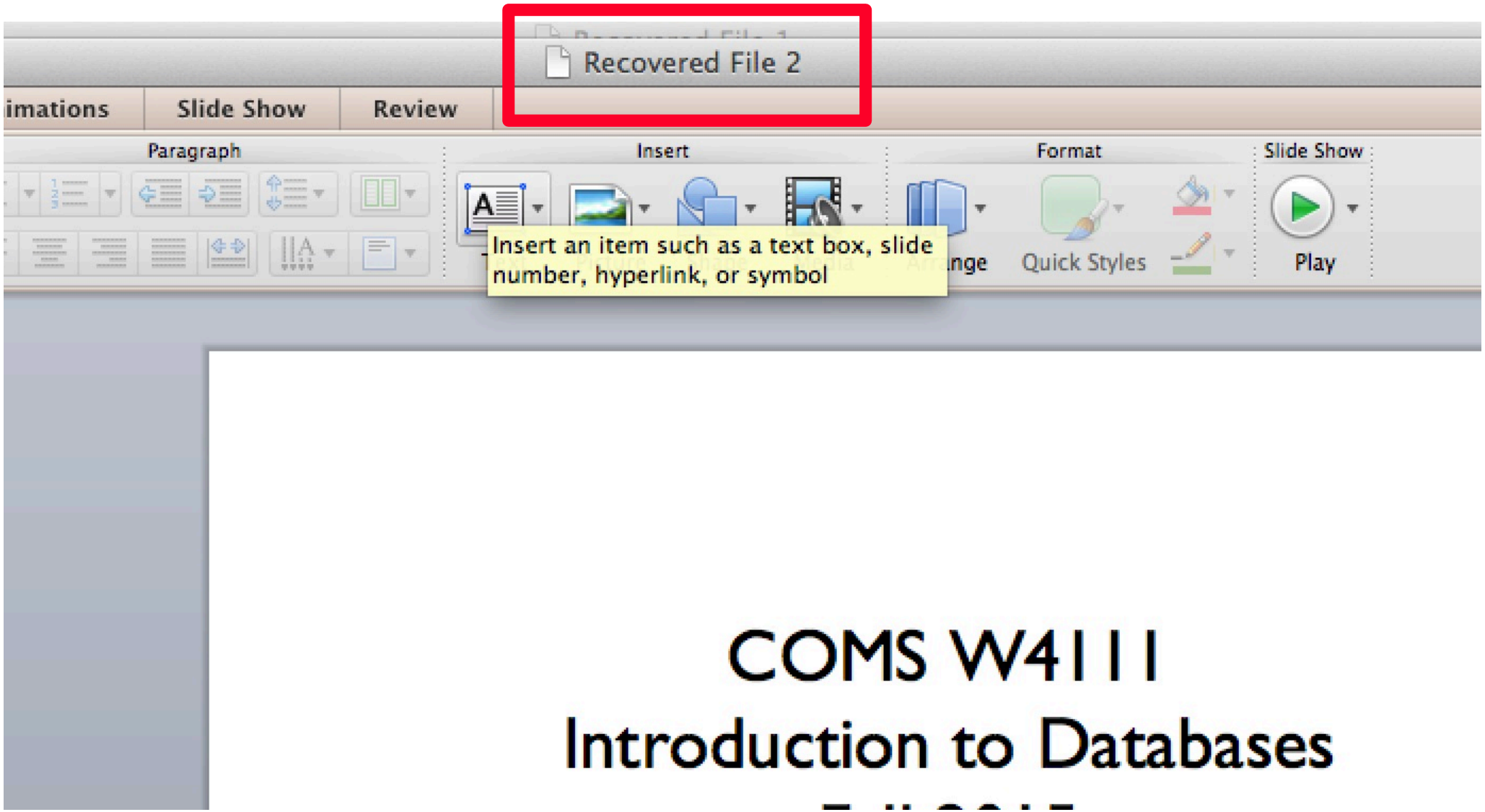
You edit a text file

Computer crashes

What happens?

1. All changes survive
2. No changes survive
3. Changes from last save survive
4. ¯_(ツ)_/¯

Is the file system a DBMS?

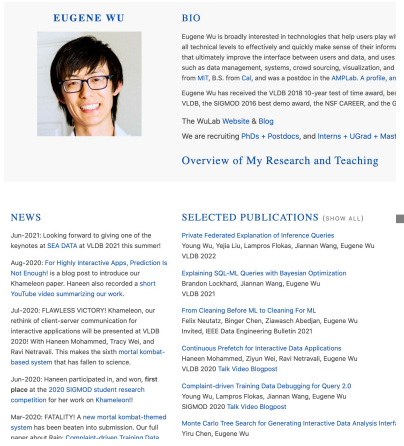


Who… would ever do this?

Real $1B+ Companies…

Store extracted data in a file

Every change → rewrite the file

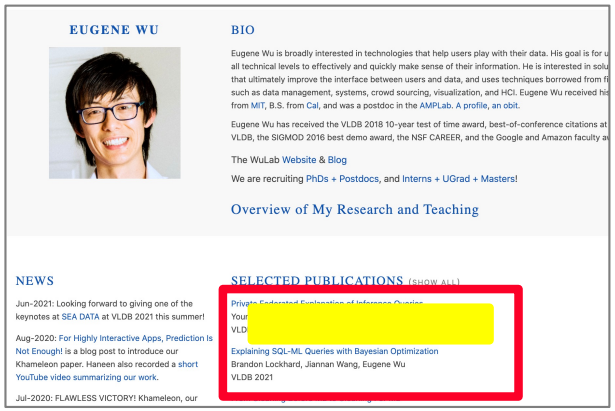


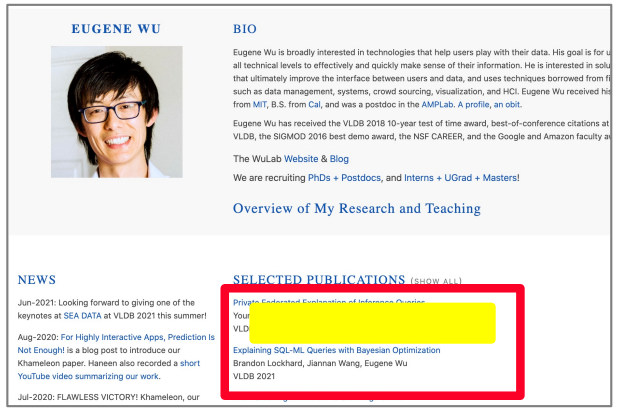
Text extraction and transform tasks

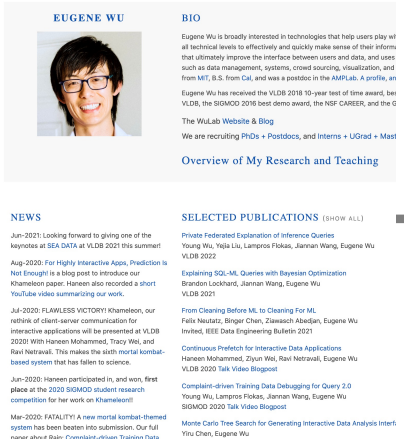
```
[{                          [{
  obj: region                 obj: authorlist
  box: 36,34,100,120          box: 40,40,100,20
  text: "Private F.."         text: "Eugene W.."
 }, { .. }, ..]              }, { .. }, ..]
```

Browser

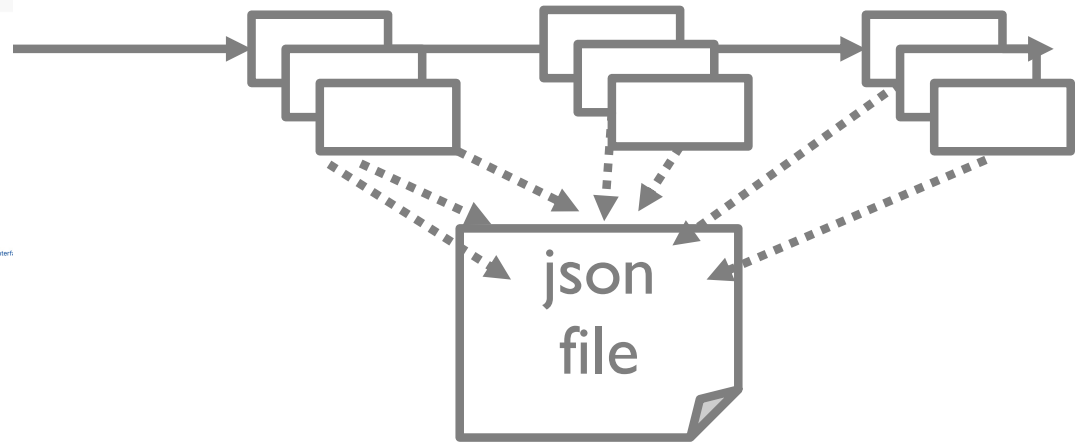
New Tab







Annotation tasks



Want *Guarantees* from DBMS

You want to write a hot new app on a DBMS.
What do you *not* want to worry about?

Failures disk, machine, human, corruption, deity

Lots of users concurrency, scaling, responsiveness

Ad-hoc data access arbitrary queries

Data formats csv? tsv? custom format?

Database Management System (DBMS)

System to safely and reliably store lots of persistent structured data and is convenient for multiple users to efficiently access and modify.

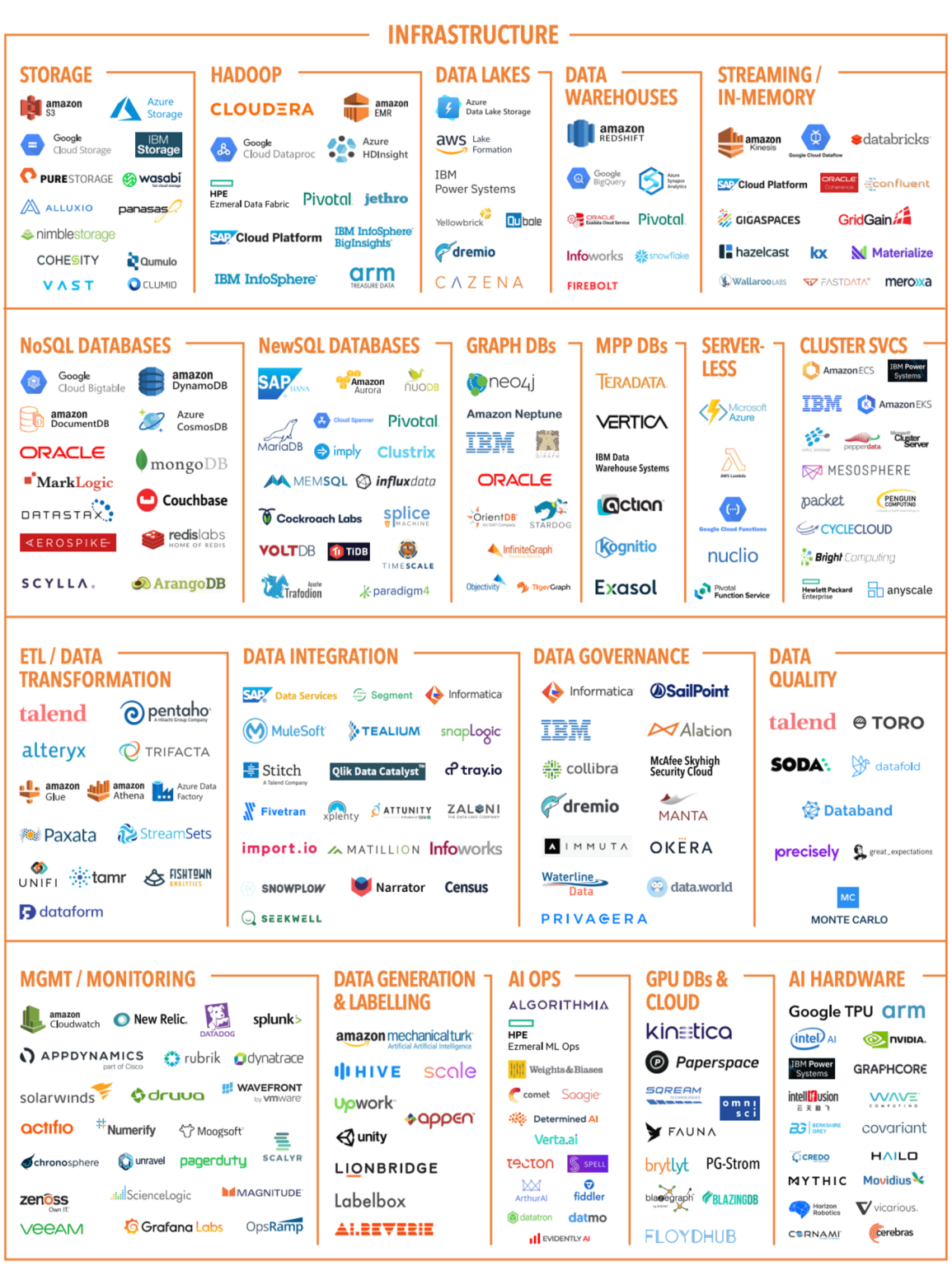
Database Management System (DBMS)

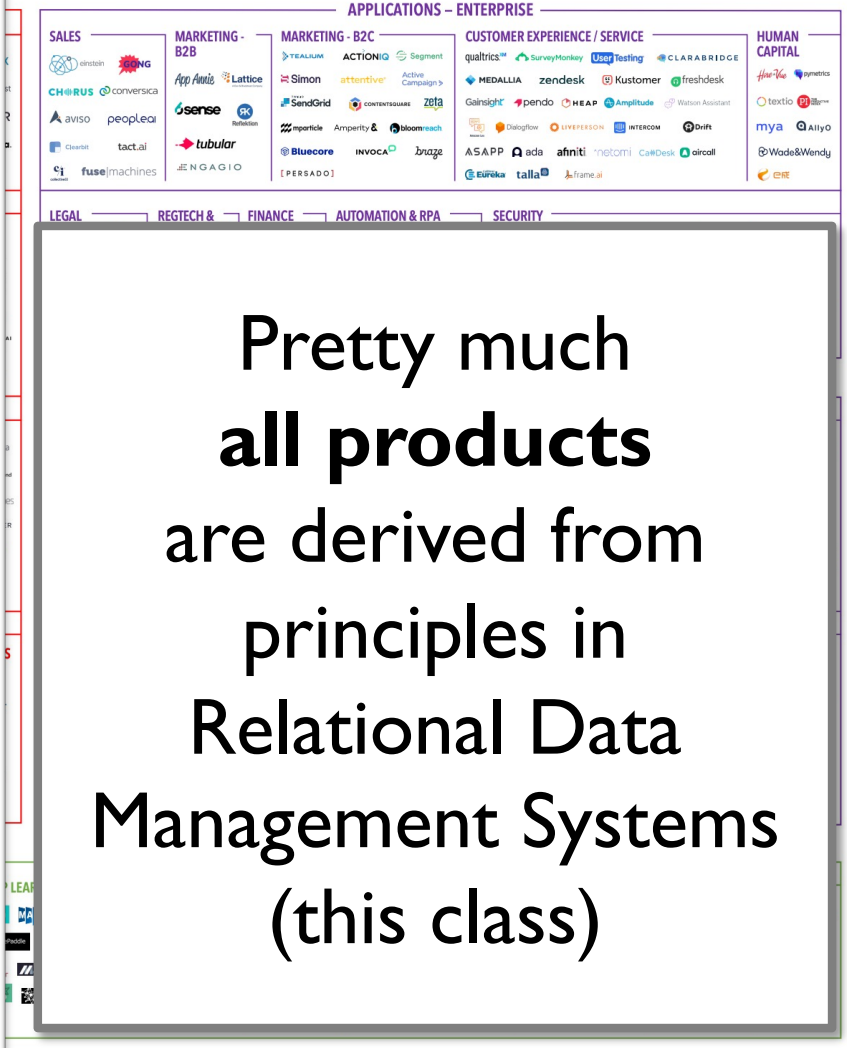
Safe	Consistent and correct data after failures
Reliable	99.99+% Uptime
Lots	>>RAM (terabytes)
Persistent	Lives longer than DBMS application
Convenient	Physical Independence. Declarative.
Multiple Users	Concurrent access. Access control.
Efficient	*Fast*: 100k+ queries / sec

Encompasses most of CS

OS	DBMS directly manages hardware
Languages	SQL is a domain specific language
Theory	Algorithms, models, NP-complete
AI/ML	Knowledge Discovery, KDD
Logic	Relational Algebra = 1st order logic

Scalable Computer Science





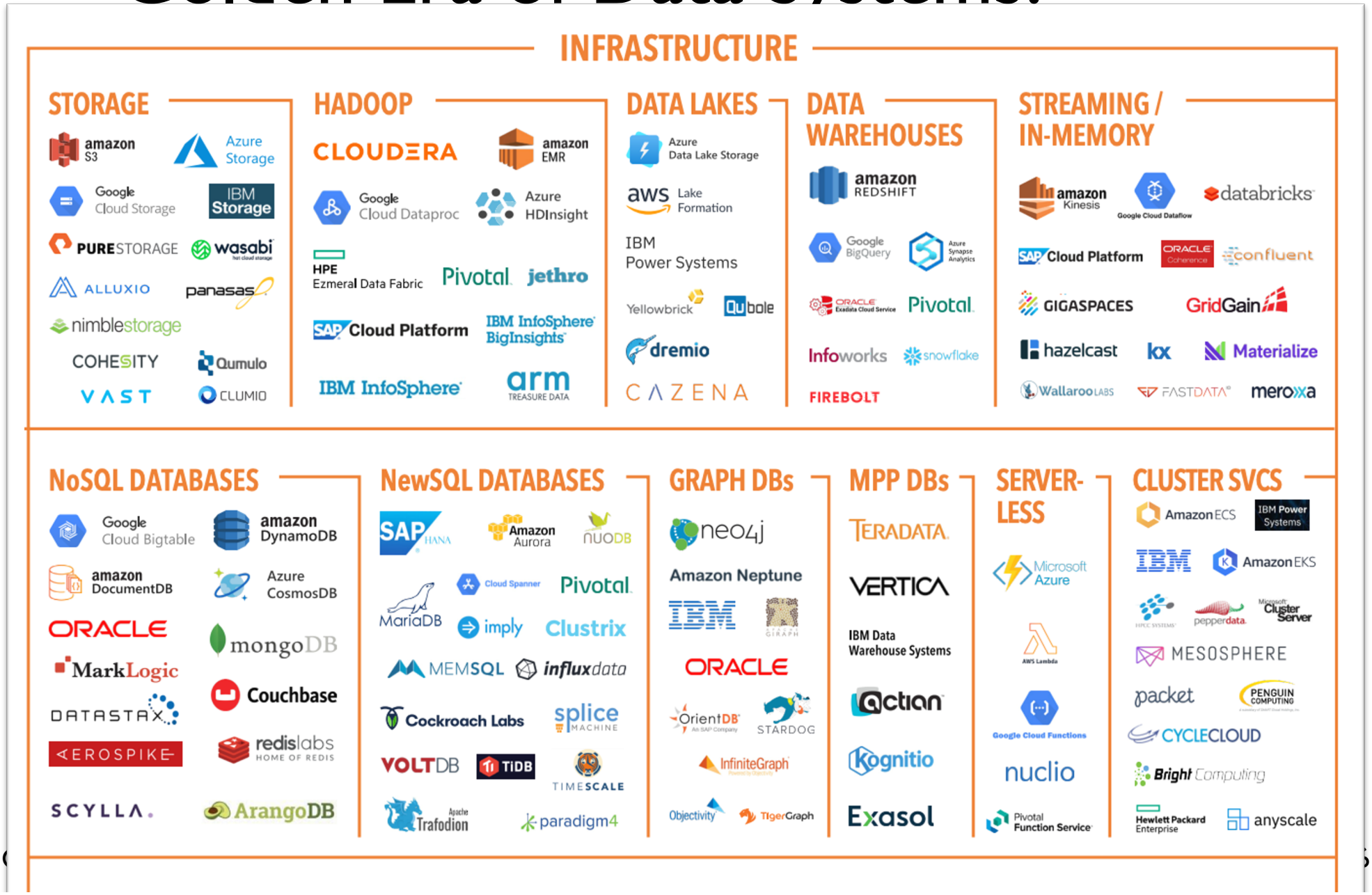
Pretty much
all products
are derived from
principles in
Relational Data
Management Systems
(this class)

tturck.com/data2020

FIRSTMARK
EARLY STAGE VENTURE CAPITAL

https://mattturck.com/data2020/

Golden Era of Data Systems!



2 Key Concepts

Data Independence

Declarative Languages

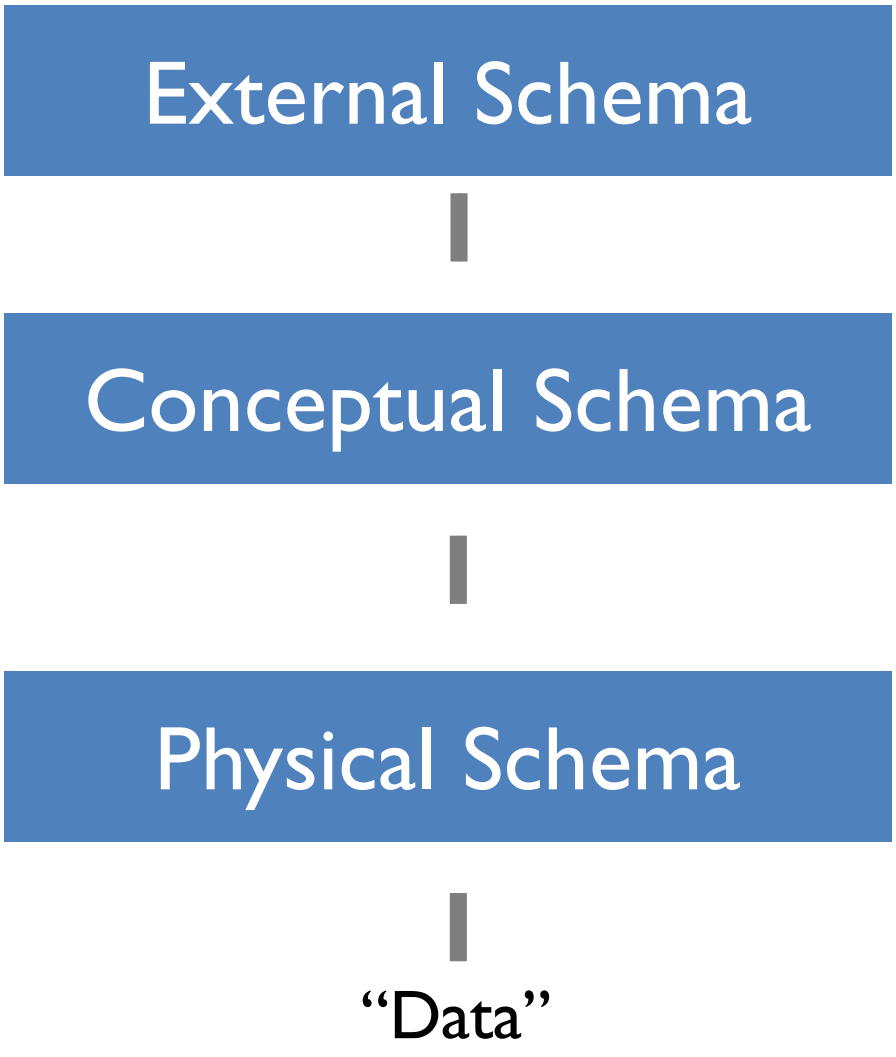
Serve to insulate application programmers from the system implementation

Data Independence

External Schema
Describe how users see data

Conceptual Schema
Describes logical structure

Physical Schema
Describes files, formats, indexes



Example App: Guuber

Users(uid int, name str, age int)

Drivers(did int, name str)

Rides(uid int, did int, distance float, drive_time float)

Data Independence

UID	Name	Age
0	Eugene	17
1	Luis	20
2	Ken	30

0,Eugene,17
1,Luis,20
2,Ken,30

CSV File

What is the number of adults?

Data Independence

UID	Name	Age
0	Eugene	17
1	Luis	20
2	Ken	30

0,Eugene,17
1,Luis,20
2,Ken,30

CSV File

```
n = 0
for line in csv_file:
  attributes = line.split(“,”)
  if attributes[2] >= 18:
    n += 1
```

Data Independence

UID	Name	Age
0	Eugene	17
1	Luis	20
2	Ken	30

0 Eugene 17
1 Luis 20
2 Ken 30

TSV File

```
n = 0
for line in csv_file:
  attributes = line.split(“,”)
  if attributes[2] >= 18:
    n += 1
```

Data Independence

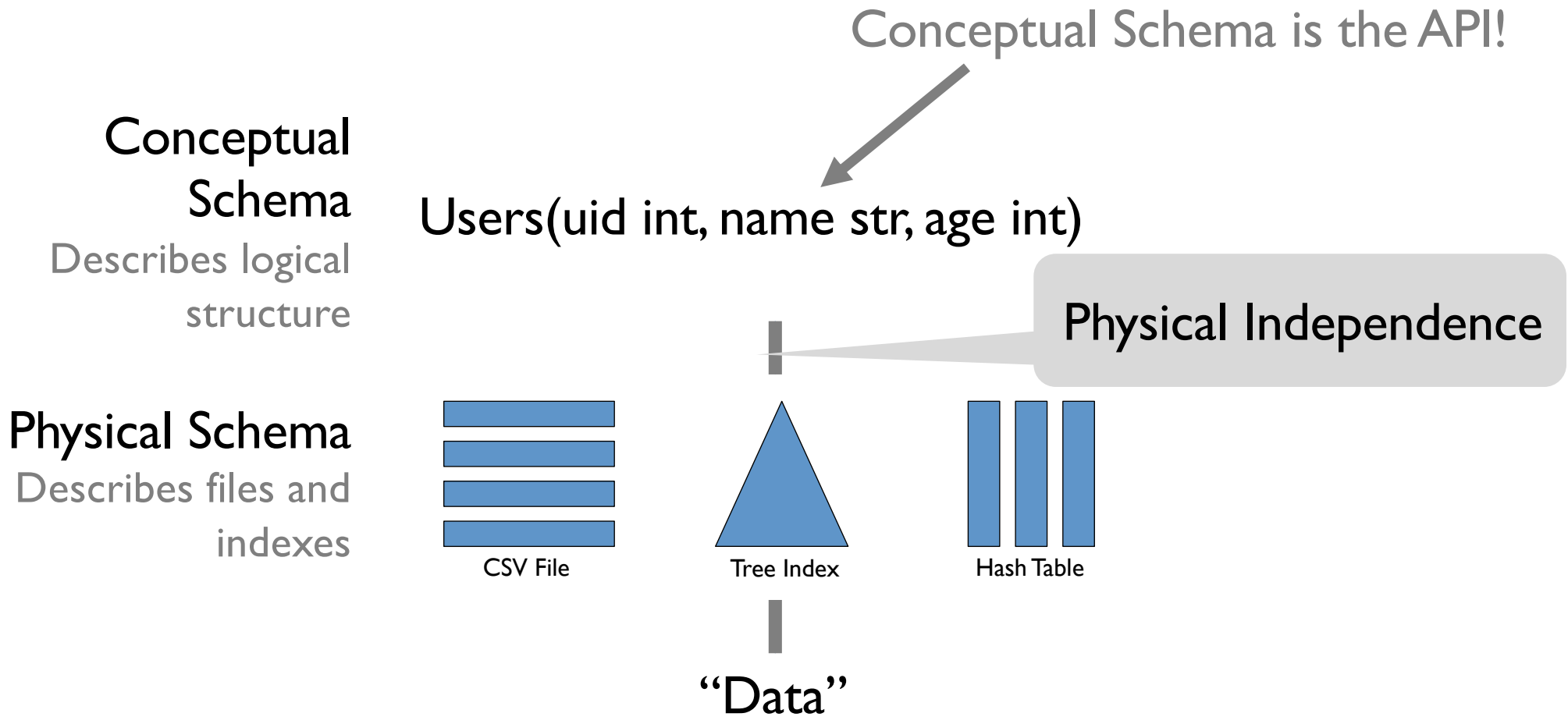
UID	Name	Age
0	Eugene	17
1	Luis	20
2	Ken	30

0,1,2
Eugene,Luis,Ken
17,20,30

Columnar File

```
n = 0
For line in csv_file:
  attributes = line.split(“,”)
  if attributes[2] >= 18:
    n += 1
```

Data Independence



Data Independence

Users(uid int, name str, age int)

"Welcome back Mr. Wu"

Data Independence

Users(uid int, fname str, lname str, age int)

"Welcome back Mr. Wu"

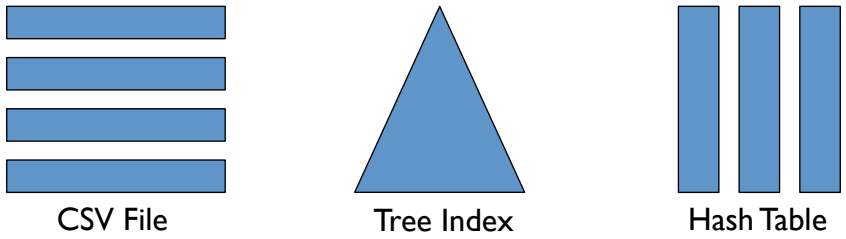
Data Independence

Conceptual Schema
Describes logical structure

Users(uid int, name str, age int)

Physical Independence

Physical Schema
Describes files and indexes



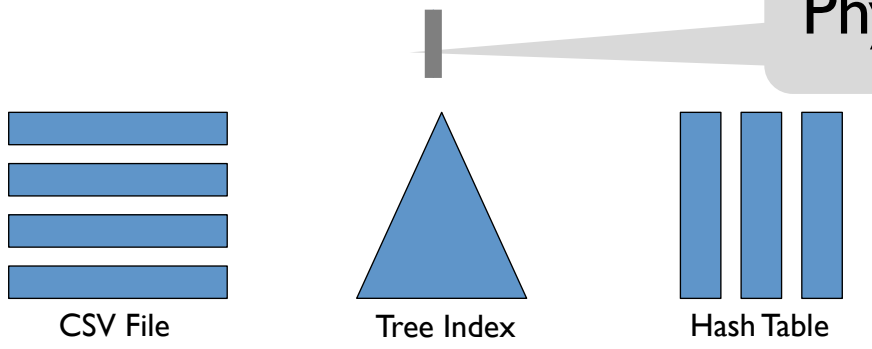
"Data"

Data Independence

Conceptual Schema
Describes logical structure

Users(uid int, fname str, lname str, age int)

Physical Schema
Describes files and indexes



Physical Independence

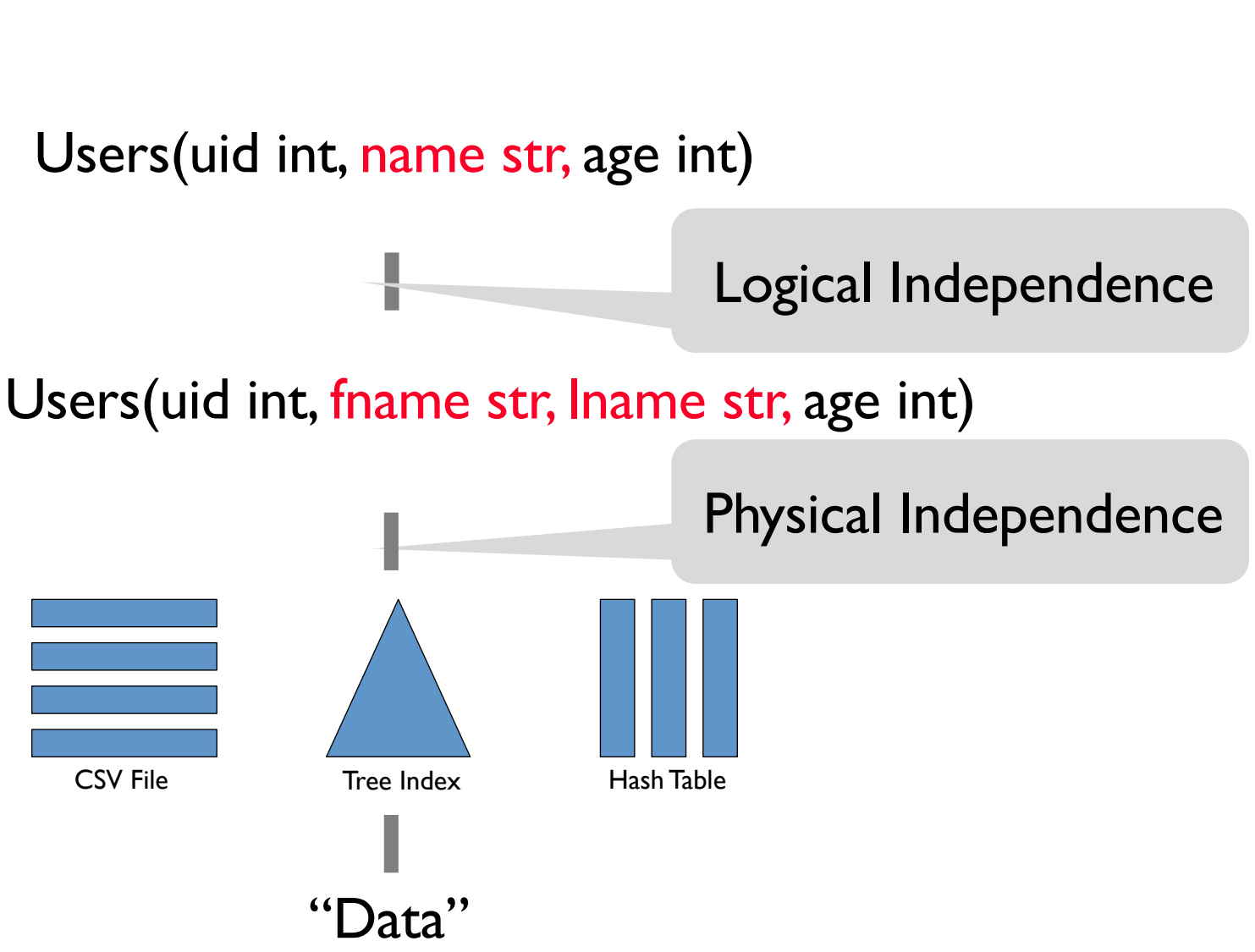
"Data"

Data Independence

External Schema
Describe how users see data

Conceptual Schema
Describes logical structure

Physical Schema
Describes files and indexes



Data Independence

Physical Independence

Protection from changes in physical structure of data

Logical Independence

Protection from changes in logical structure of data

One of most important properties of a DBMS

Declarative Interface

Mechanism that enables data independence

Insulates programmer from physical schema

Rather than a list of functions,

the API is a *query language*

Declarative Interface

What you want, **not how to do it.**

"Make me a sandwich"

Buy from pb&j store
Make BLT
½ Tuna
Veggie

"Take two slices of wheat bread out of the 2nd shelf, put them next to each other…"

What if on 1st shelf?
Out of wheat bread?
No counter space?

Declarative Interface

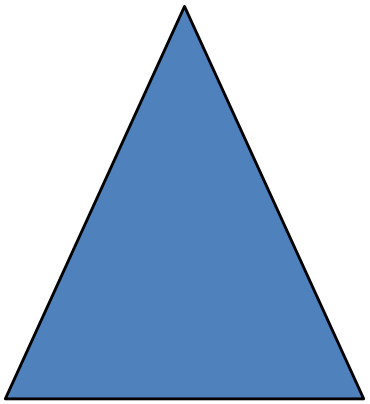
"I want all highly rated fast drivers"

Declarative Interface

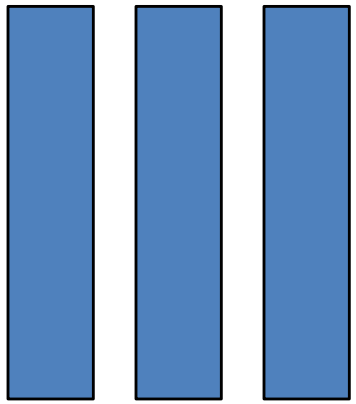
SELECT name FROM users WHERE rating > 8

DBMS

CSV File



Tree Index

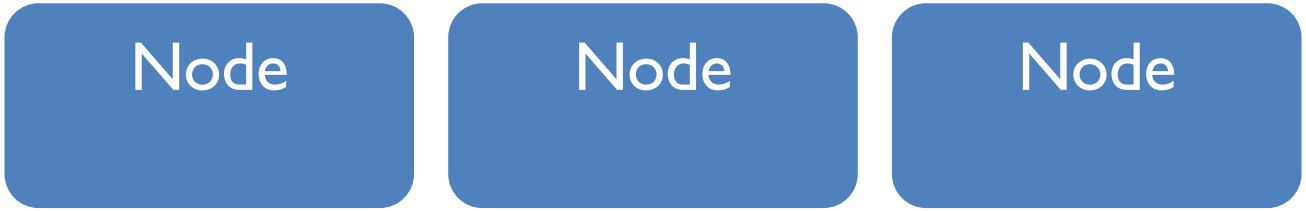


Hash Table

Declarative Interface

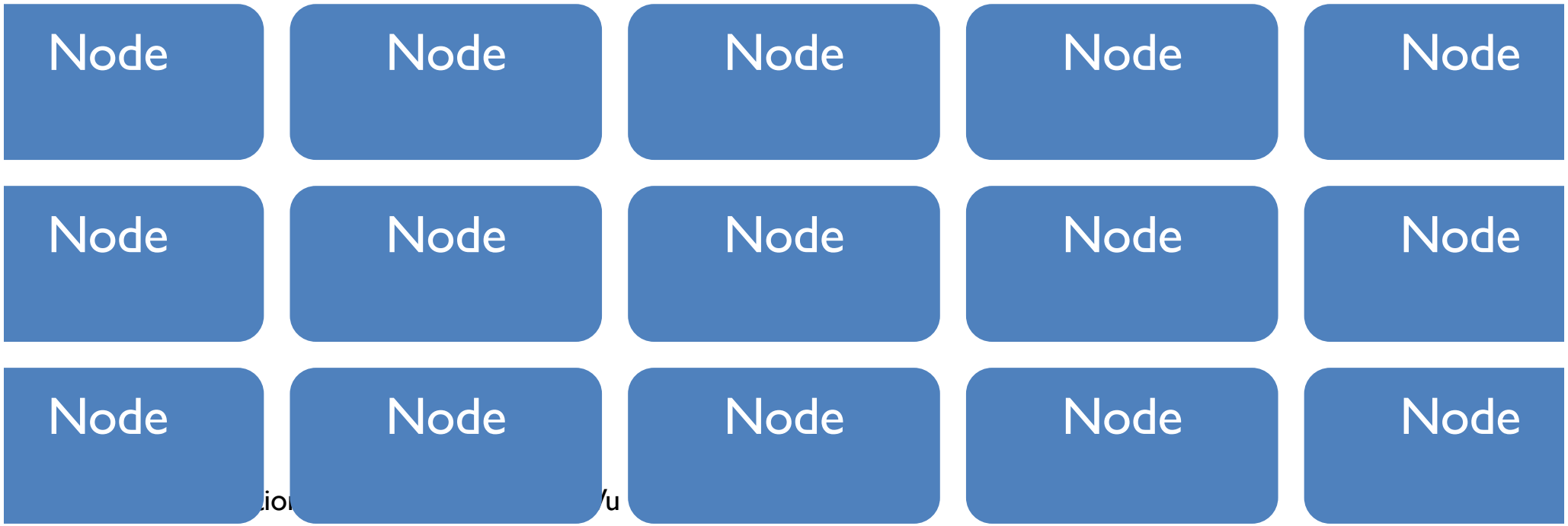
SELECT name FROM users WHERE rating > 8

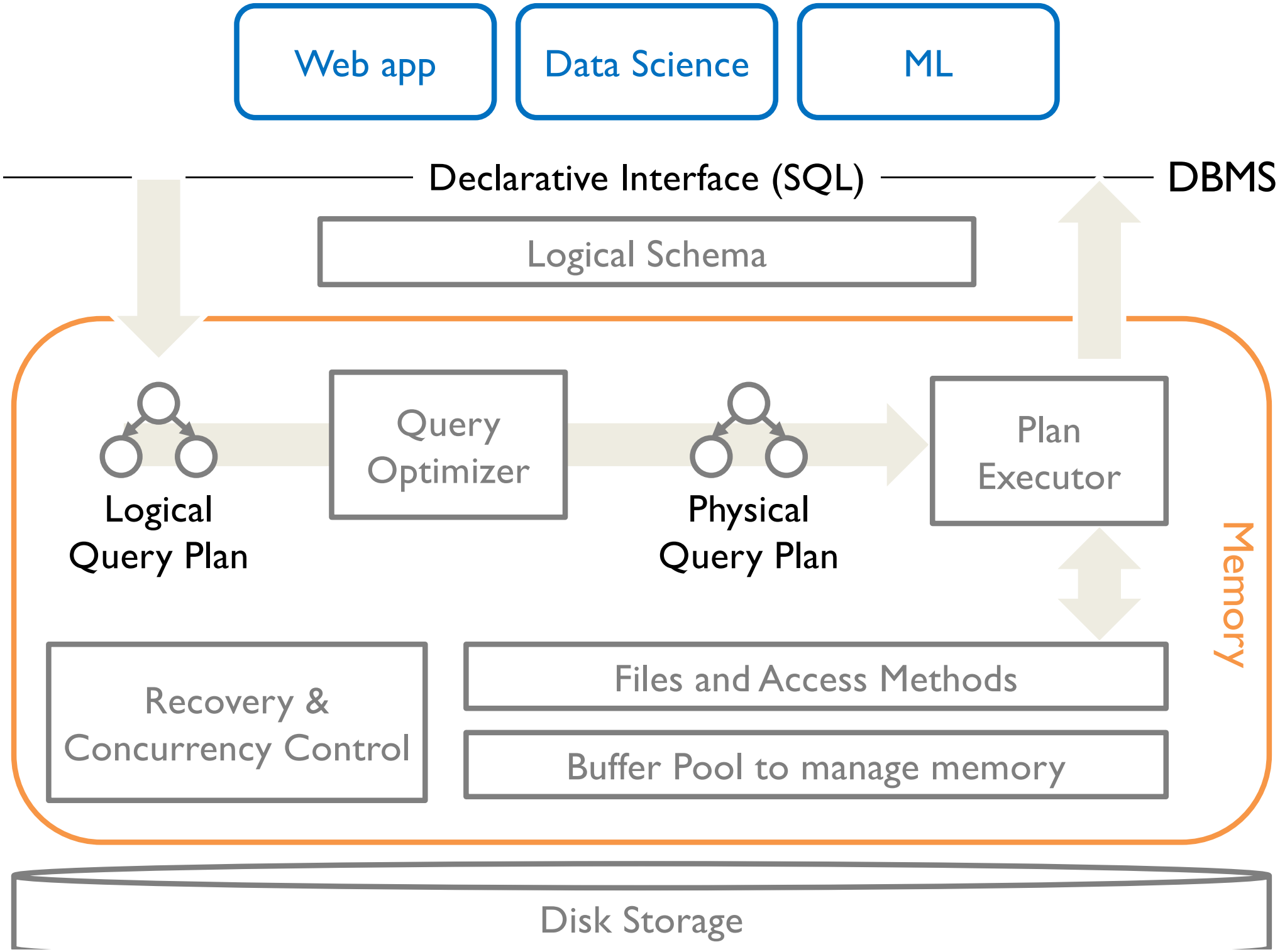
DBMS

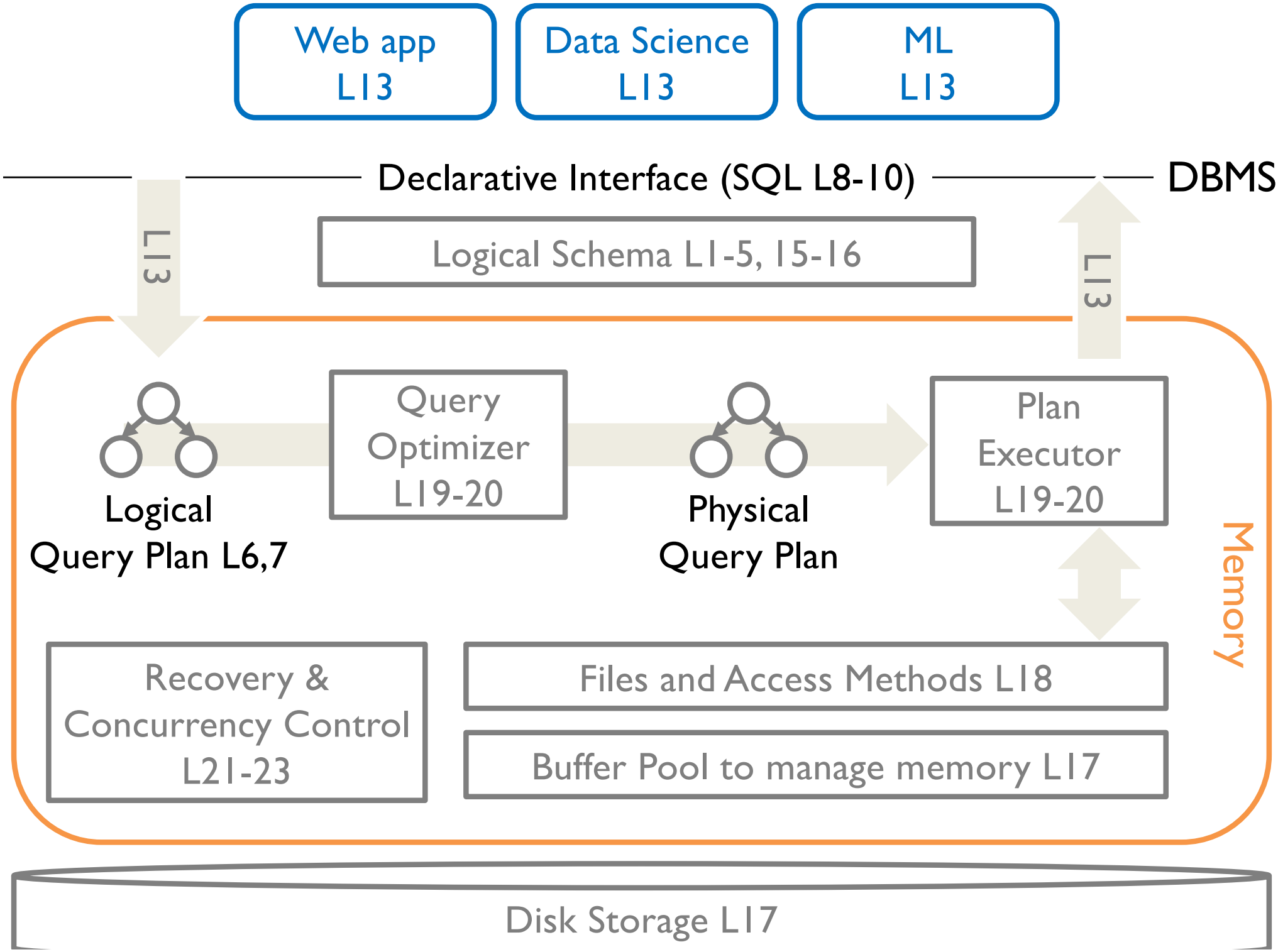


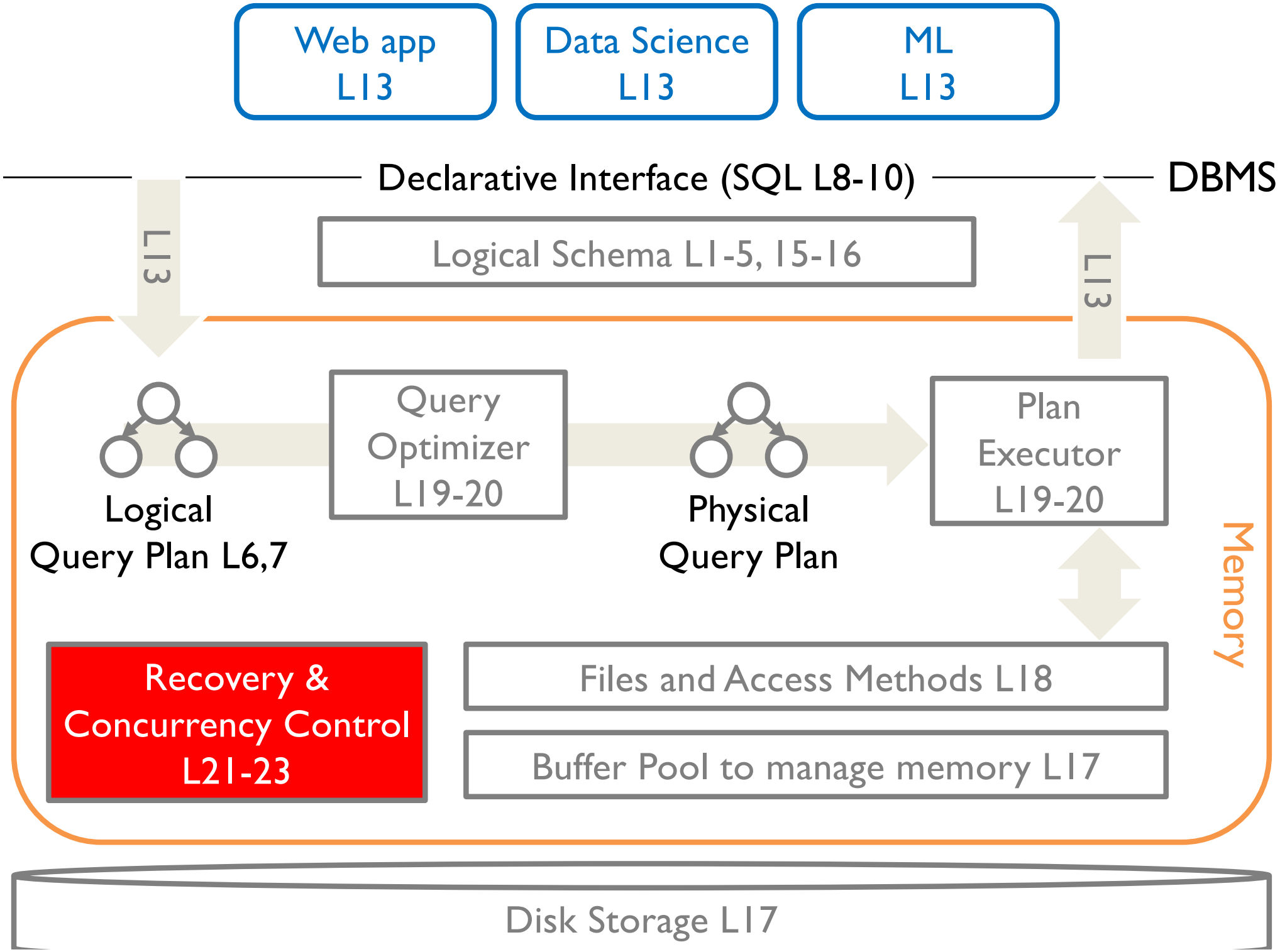
Declarative Interface

SELECT name FROM users WHERE rating > 8




Web app
Data Science
ML
Declarative Interface (SQL)
DBMS
Logical Schema
Query Optimizer
Logical Query Plan
Physical Query Plan
Plan Executor
Memory
Recovery & Concurrency Control
Files and Access Methods
Buffer Pool to manage memory
Disk Storage


Web app
L13
Data Science
L13
ML
L13
Declarative Interface (SQL L8-10)
DBMS
L13
Logical Schema L1-5, 15-16
L13
Query
Optimizer
L19-20
Plan
Executor
L19-20
Logical
Query Plan L6,7
Physical
Query Plan
Memory
Recovery &
Concurrency Control
L21-23
Files and Access Methods L18
Buffer Pool to manage memory L17
Disk Storage L17


Web app
L13
Data Science
L13
ML
L13
Declarative Interface (SQL L8-10)
DBMS
L13
Logical Schema L1-5, 15-16
L13
Query
Optimizer
L19-20
Plan
Executor
L19-20
Logical
Query Plan L6,7
Physical
Query Plan
Memory
Recovery &
Concurrency Control
L21-23
Files and Access Methods L18
Buffer Pool to manage memory L17
Disk Storage L17

Concurrency Control

Want to let many users use database concurrently.

How to ensure they run correctly?

Concurrency Control

Want to let many users **use** database concurrently.

How to ensure they run correctly?

What does "**use**" mean?

Run transaction, which groups all of the user's DBMS actions together. They either all run, or none run.

```
Begin;
<read beth's account>
<deduct from beth's account>
<increase eugene's account>
Commit;  (or Abort;)
```

Concurrency Control

Want to let many users use database concurrently.

How to ensure they run **correctly**?

What does "**Correctly**" mean?

Data in DBMS is correct if

1. Maintains all integrity constraints (types, references, checks)
2. Transactions run as if only one DBMS user at a time.

Recovery

If the DBMS crashes, can recover to a **Correct** state.

What does "**Correct**" mean?

All committed transactions are preserved in the database.

Undo all incomplete (uncommitted) transactions.

How?

DBMS keeps a **log** of all actions each transaction performs.

Track whether an action is committed or uncommitted.

A bit about the class

Next Up

HW0 is out.

https://github.com/w4111/hw0

Due by 9/11 11:59PM.

No late submissions accepted

Class Information: Prerequisites

COMS W3134 - *Data Structures in Java* or
COMS W3137 - *Data Structures and Algorithms*

(equivalent courses taken elsewhere are acceptable as well)

Fluency in Python

Class Information: Lectures

Tu/Th

10-11:30AM

451 CSB

Your TAs

Office hours will be updated on course website later today.

Zoom links for office hours will be shared on the discussion board

w4111.github.io

COLUMBIA UNIVERSITY COMS W4111

INTRODUCTION TO DATABASES

Information

- Tues/Thurs 10-11:30
  451 CSB
  3 units
- Syllabus
- Ed Discussion
- Provide Feedback
- Course Github

Staff

- Eugene Wu Instructor
  Thurs 12-1PM
- Zachary Huang
  tba
- Andrew Zheng
  tba
- Jennifer Wang
  tba
-
  tba
-
  tba

Office Hours

- OH Links
- OH Calendar

Prereqs

- Required: Students are expected to be comfortable with data

Overview

The goal of this class is two-fold. First, to introduce you to core database concepts (e.g., data modeling, logical design, SQL) so that you too can build a billion dollar application. Second, to teach enough about database engine internals (e.g., physical database design, query optimization, transaction processing) so you have a good sense of why queries may be running slowly/incorrectly. We will also discuss their relevance to systems used in industry.

The Data Management Seminar invites interesting database researchers and practitioners to speak. Students are invited to join in person or on zoom (if available). We will announce these periodically throughout the semester.

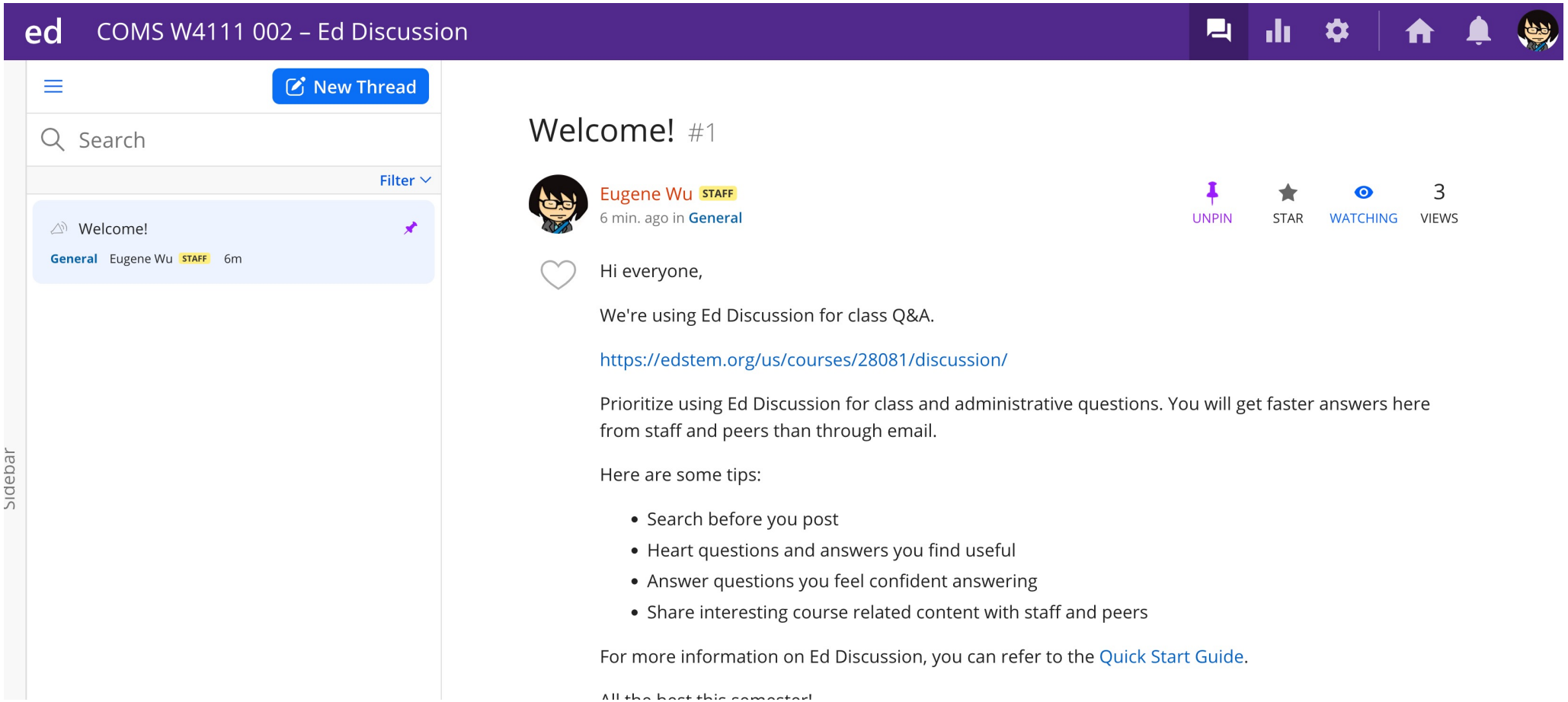
Announcements

- Sign up for Project 1 Part 1 staff meetings! One meeting per team.
- Updated lecture 2 slides to clarify constraints over N-way relationships.
- HW0 released. No Late Days! Failure to submit on time is a -5% penalty on your final grade.

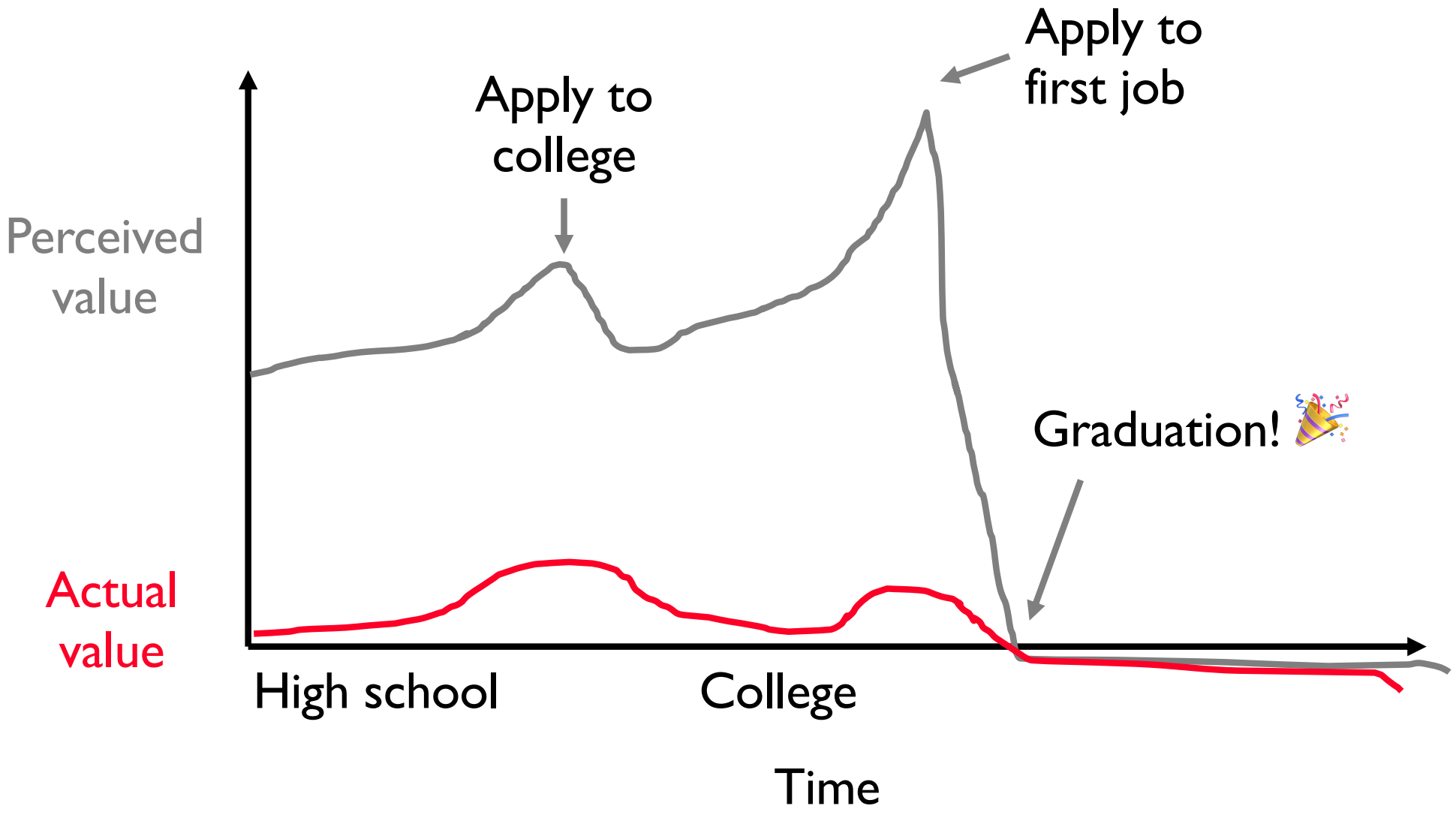
Schedule

Date	Topic	Assigned	Due
6-Sep	**Intro and Overview**	HW 0 Look for teammates	
8-Sep	**ER Models** optional: Textbook Chapter 6 except for Sections 6.7, 6.10, and 6.11.	HW1 Part 1 Project 1 Part 1	HW0 (9/11 11:59PM EST. NO LATE DAYS)
13-Sep	**ER Models** optional: Textbook Chapter 6 except for Sections 6.7, 6.10, and 6.11.		HW 1 Part 1 (9/16 11:59PM EST) Formed Project 1 Team (no submission)
15-Sep	**Relational Model** optional: Textbook Ch 2.1-2.3, 2.5, 6.7, 6.8, except 6.7.2		Project 1 Part 1 approval phase
20-Sep	**Relational Model** optional: Textbook Ch 2.1-2.3, 2.5, 6.7, 6.8, except 6.7.2	HW1 Part 2 Schema	Project 1 Part 1 approval phase
22-Sep	**Relational Algebra**		Project 1 Part 1 approval phase

Discussion Board



Grades. How do they work?



Grading Information

Midterm 1	25%
Midterm 2	40%
HW	15% (4 HWs equally weighed)
Project 1	15%
Project 2	5%
Extra credit	variable

Median grade: B or slightly higher.

Exam Dates

Midterm 1	10/13	on gradescope
Midterm 2	12/8	on gradescope, cumulative

Makeup exams are not scheduled

Homework

Assignment will specify submission instructions.

No extensions or exceptions.

5 grace days for hws throughout the semester.

Can be applied to any assignment *unless otherwise specified*

After using all grace days, 25% grade deduction per day.

Don't need to tell us, staff will assign grace days in your favor

Check full details on web site under syllabus.

Projects (more details soon)

Two projects.

- Teams of two
- Run on cloud infrastructure
- Python & SQL

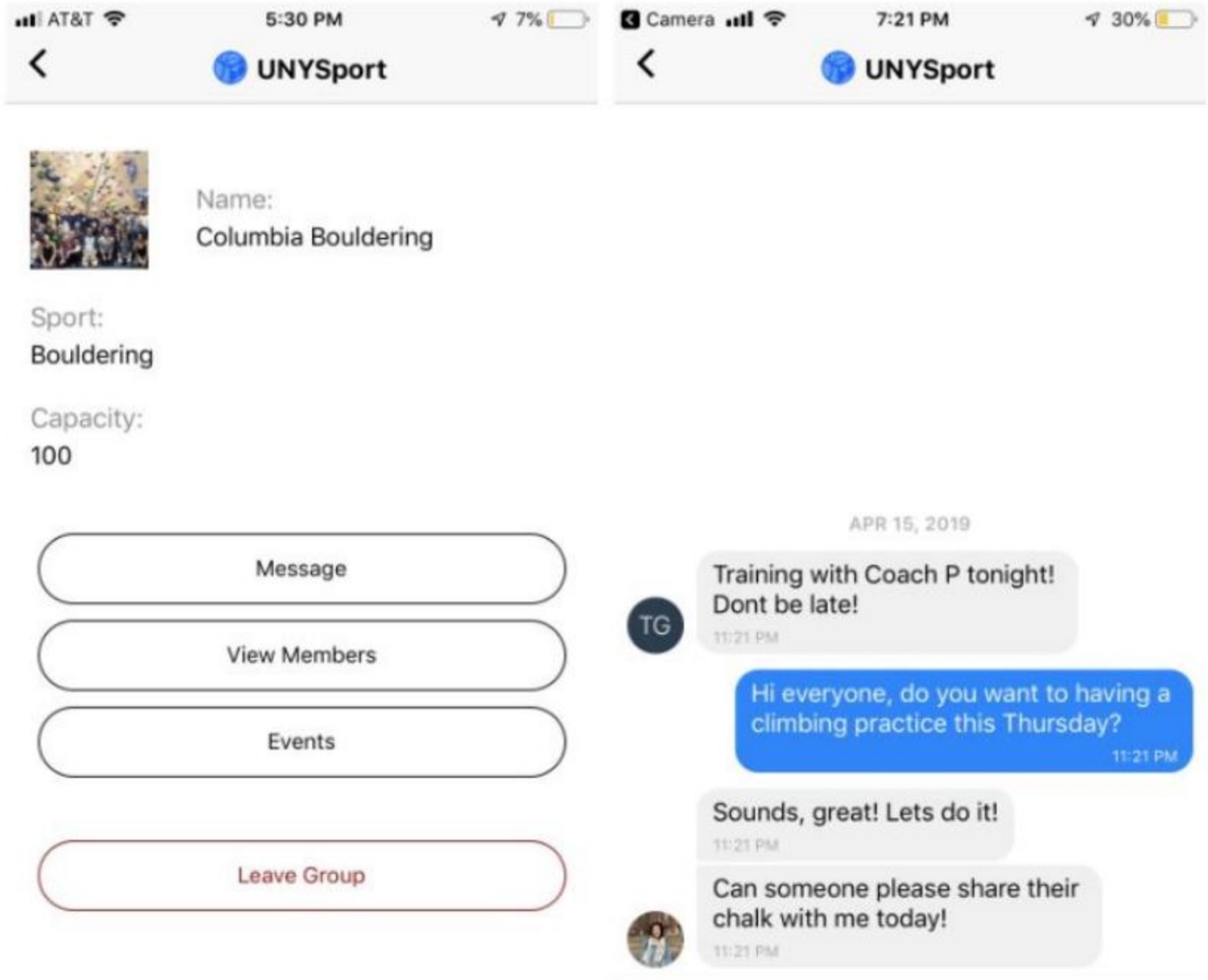
Project 1

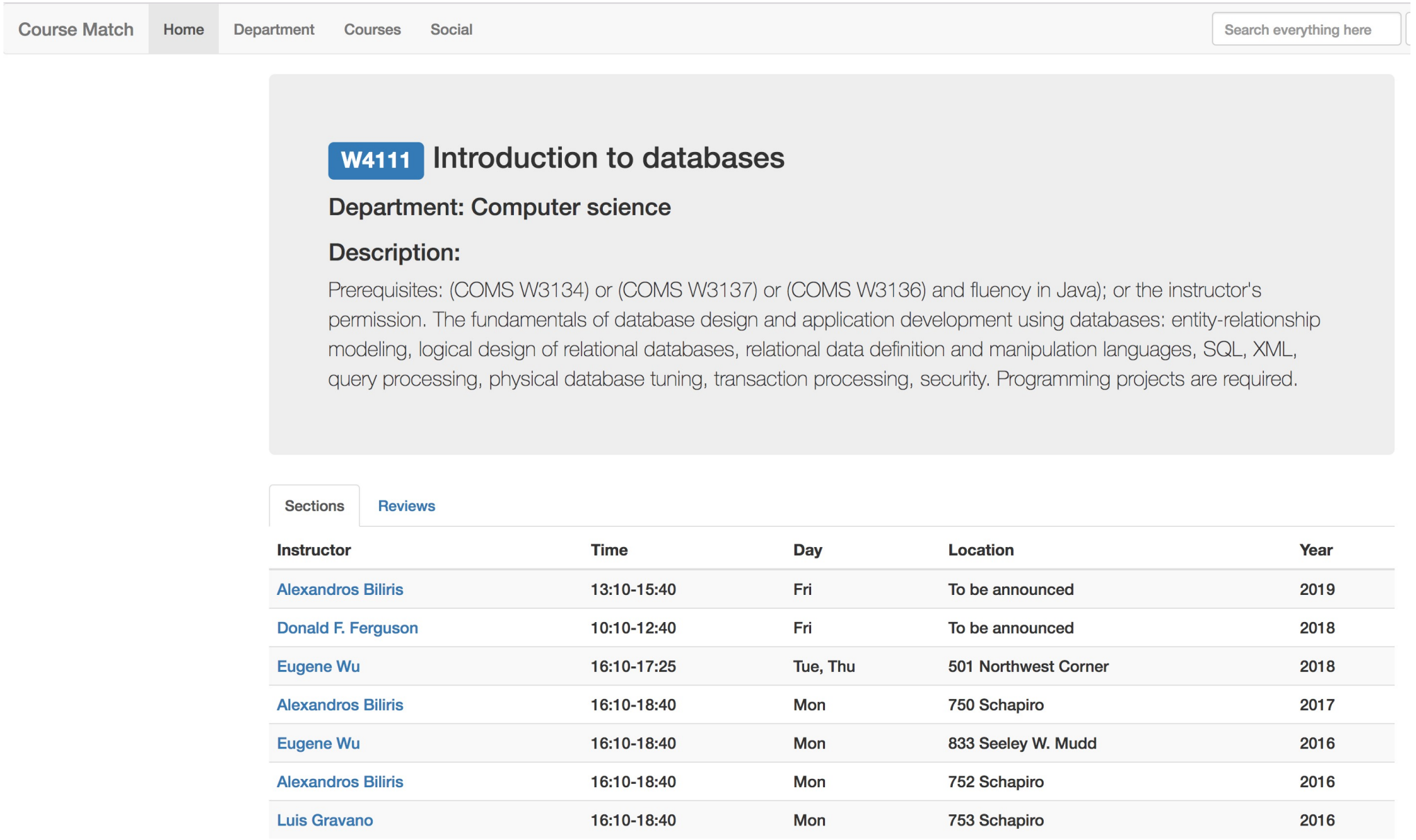
- Model and build your own database web application
- Explore "traditional" relational database features.
- Non-programming option

Project 2

- Do cool things with DBMSes

Sports Community Mobile App



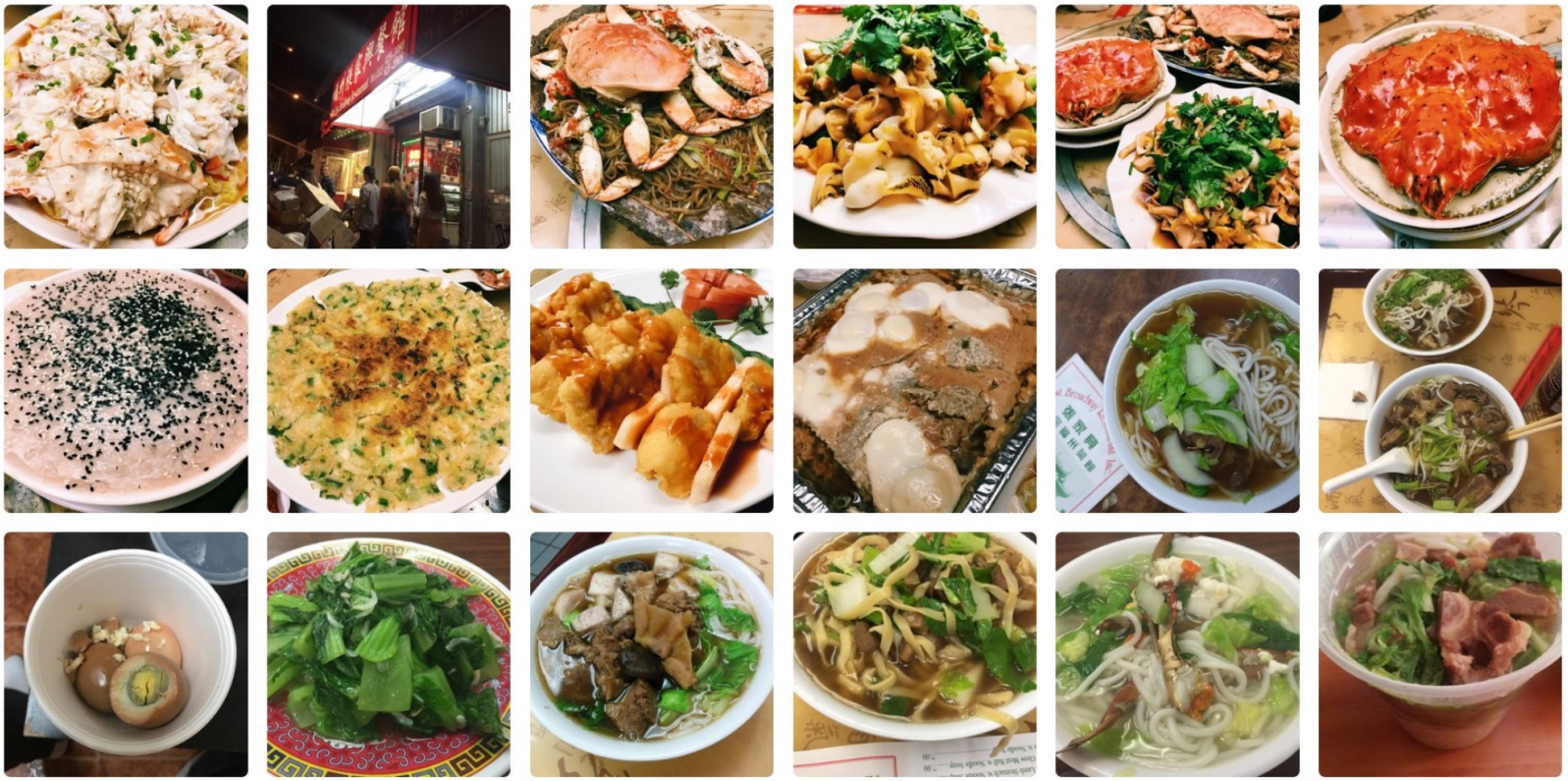

Course Match
Home
Department
Courses
Social
Search everything here
W4111 Introduction to databases
Department: Computer science
Description:
Prerequisites: (COMS W3134) or (COMS W3137) or (COMS W3136) and fluency in Java); or the instructor's permission. The fundamentals of database design and application development using databases: entity-relationship modeling, logical design of relational databases, relational data definition and manipulation languages, SQL, XML, query processing, physical database tuning, transaction processing, security. Programming projects are required.
Sections
Reviews
Instructor Time Day Location Year
Alexandros Biliris 13:10-15:40 Fri To be announced 2019
Donald F. Ferguson 10:10-12:40 Fri To be announced 2018
Eugene Wu 16:10-17:25 Tue, Thu 501 Northwest Corner 2018
Alexandros Biliris 16:10-18:40 Mon 750 Schapiro 2017
Eugene Wu 16:10-18:40 Mon 833 Seeley W. Mudd 2016
Alexandros Biliris 16:10-18:40 Mon 752 Schapiro 2016
Luis Gravano 16:10-18:40 Mon 753 Schapiro 2016

C-Food

Reviews Complaints Register Log In Log Out

C-Food: Your guide to clean NYC Restaurants

East Broadway Restaurant



Borough: manhattan

Address: 94 East Broadway

Health Investigation Score: 41/50

Average User Rating: 3.20

Domino's

Projects (cont.)

3 grace days total for project parts 1 and 2.

No extensions or exceptions for project part 3 submission.

After using all grace days, 25% grade deduction per late day.

Check full details on web site.

Extra Credit

From Midterms, Projects, standalone, …

all added *after* the curve

Does NOT affect those that don't do extra credit

Collaboration Policy

Read Syllabus on course site for allowed conduct

CS Dept academic honesty policies
http://www.cs.columbia.edu/education/honesty

We will not tolerate *any* cheating

Collaboration Policy

Discussing lectures and course material strongly encouraged

Homework and exams are *individual*. No exceptions
Any libraries or code however minor must be disclosed.

Projects are done in *teams*; no collaboration between teams.

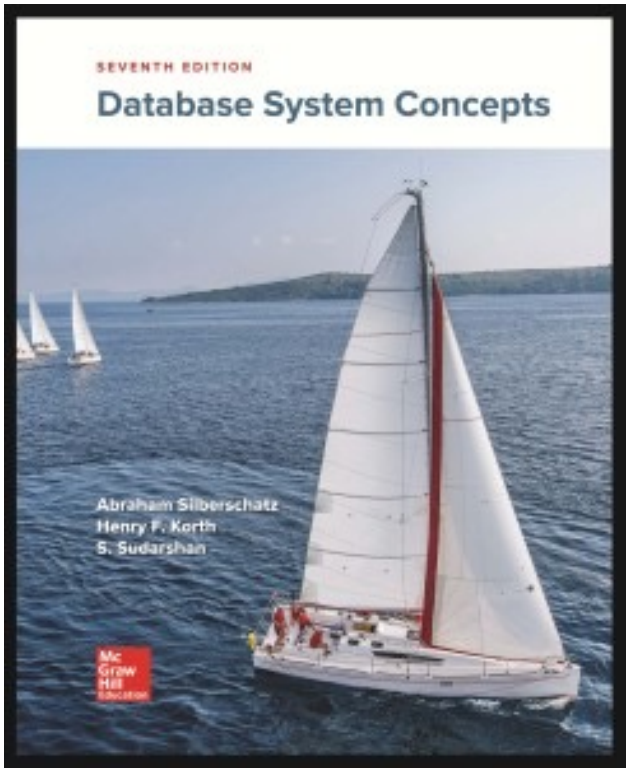
Contact the Professor Wu
right away if you have any questions or are falling behind.

Optional Textbook

Silberschatz et al.

Database System Concepts

7th ed



On-going Feedback

COLUMBIA UNIVERSITY COMS W4111

INTRODUCTION TO DATABASES

Information

- Tues/Thurs 10-11:30
  451 CSB
  3 units
- Syllabus
- Ed Discussion
- Provide Feedback
- Course Github

Staff

- Eugene Wu Instructor
  Thurs 12-1PM
- Zachary Huang
  tba
- Lia Chen
  tba
- Jessica Shi
  tba
- Jennifer Wang

Overview

The goal of this class is two-fold. First, to introduce you to core database concepts (e.g., data modeling, l
build a billion dollar application. Second, to teach enough about database engine internals (e.g., physical
transaction processing) so you have a good sense of why queries may be running slowly/incorrectly. We
systems used in industry.

nt Seminar invites interesting database researchers and practitioners to speak. Stuc
/e will announce these periodically throughout the semester.

Announcements

- HW0 released. No Late Days! Failure to submit on time is a -5% penalty on your final grade.

Schedule

	Date	Topic	Assigned
L1	6-Sep	**Intro and Overview**	HW 0 Look for teammates
L2	8-Sep	**ER Models**	HW1 Part 1

Database Courses at Columbia

COMS W4111 - Intro to Databases

Prerequisites: CS3137 or CS3134; fluency in Python

Intro to DBMSes

Data Models

Relational Algebra

SQL

Applications + SQL

Normalization

Peek at DBMS internals:

- Storage and indexing
- Query optimization
- Transaction Processing

COMS W4112-Database Sys. Impl.

Prerequisites: CS3137 or CS3134; fluency in Python

Components of a Database System in Detail

Storage Methods and Indexing

Query Processing and Optimization

Materialized Views

Transaction Processing and Recovery

Parallel & Distributed DBMSes

Performance Considerations Beyond Disk I/Os

COMS E6111-Advanced Databases

Prerequisites: CS4111; fluency in Java or Python

Information Retrieval
Information Extraction
Web Search
Data Mining
Data Warehousing, OLAP, Decision Support

COMS E6xxx-DB Research Seminars

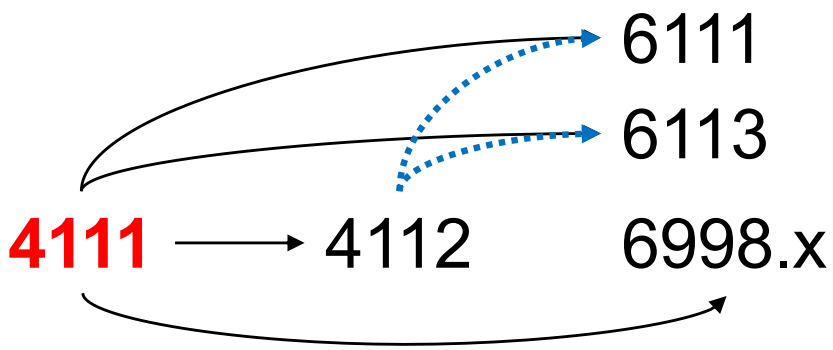
Prerequisites: CS4111; fluency in Java or Python

6113 Database Research Topics

w6113.github.io

6998.002 Systems for
Human Data Interaction

columbiaviz.github.io


6111
6113
4111
4112
6998.x

Data Management at Columbia


Luis Gravano


Kenneth Ross


Eugene Wu


Mihalis Yannakakis

`http://cudbg.github.io/`

Borrowed material from
Prof. Gravano
Prof. Hellerstein (Cal)
Prof. Madden & Stonebraker (MIT)

w4111.github.io

DO HOMEWORK 0!