Updated 9/8/2022

Topic 0: Overview
L01: Course Overview

Wolfgang Gatterbauer

CS3200 Database design (fa22)

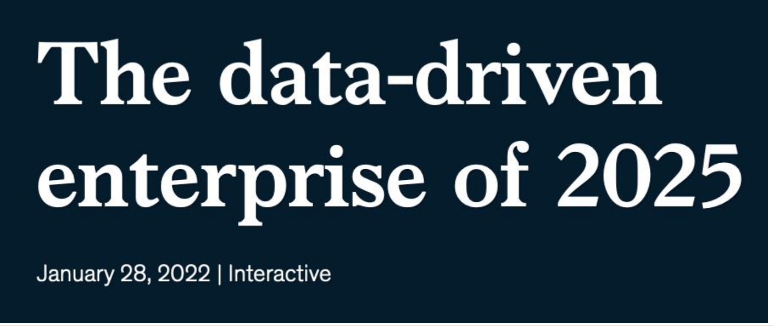
https://northeastern-datalab.github.io/cs3200/fa22s3/

9/7/2022

The world is increasingly driven by data…

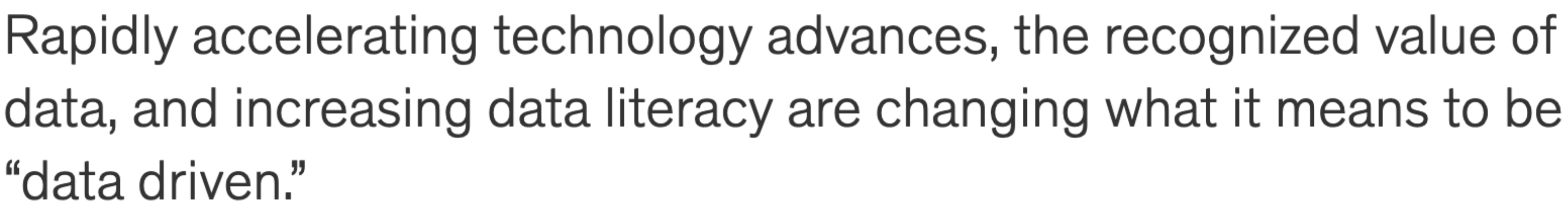
This class teaches the basics of how to manage relational data.

Increasingly, many companies see themselves as data driven. Not just the big players ...

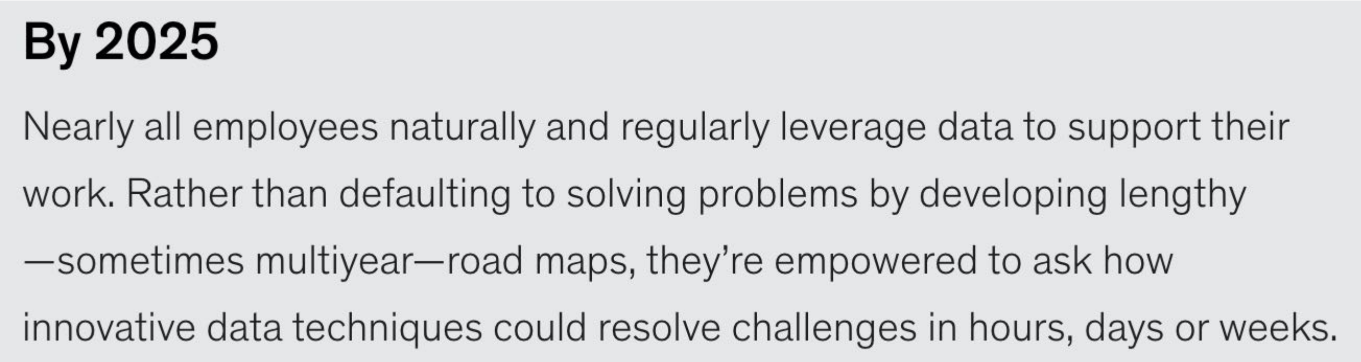


The data-driven enterprise of 2025

January 28, 2022 | Interactive

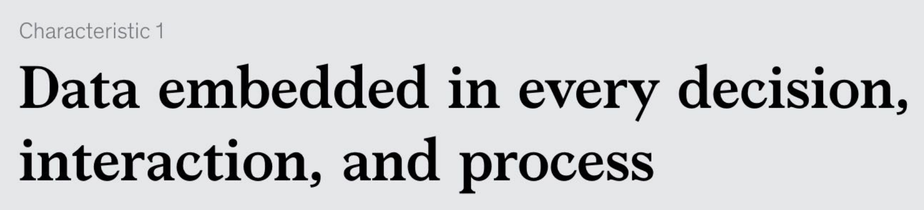


Rapidly accelerating technology advances, the recognized value of data, and increasing data literacy are changing what it means to be “data driven.”



By 2025

Nearly all employees naturally and regularly leverage data to support their work. Rather than defaulting to solving problems by developing lengthy—sometimes multiyear—road maps, they’re empowered to ask how innovative data techniques could resolve challenges in hours, days or weeks.



Characteristic 1

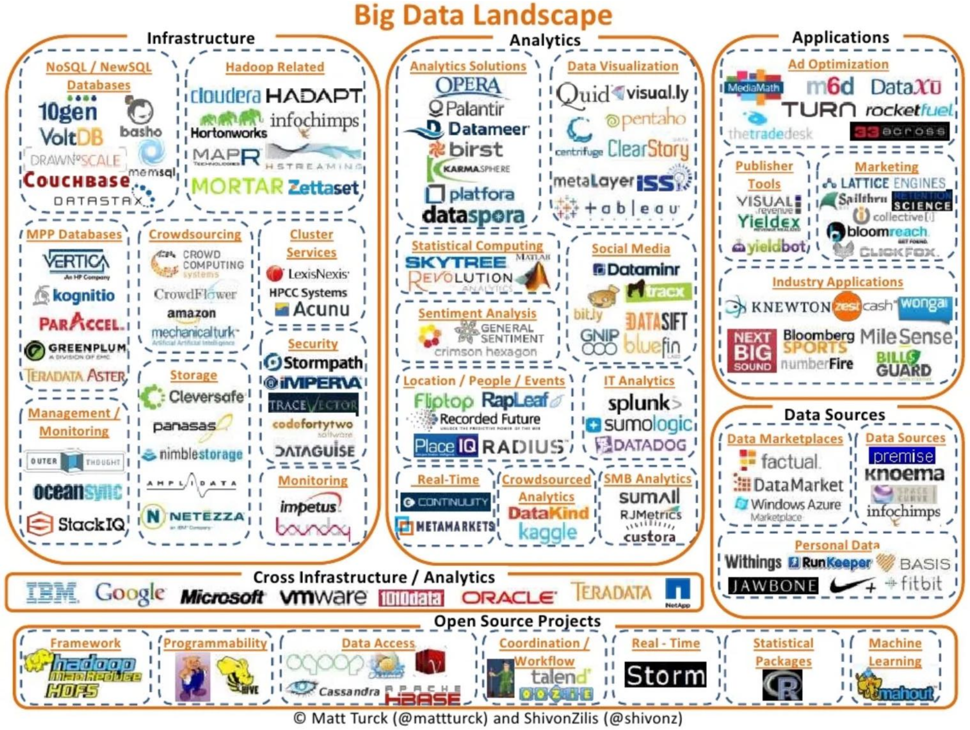
Data embedded in every decision, interaction, and process

Increasingly, many companies see themselves as <u>data driven</u>. Not just the big players ...

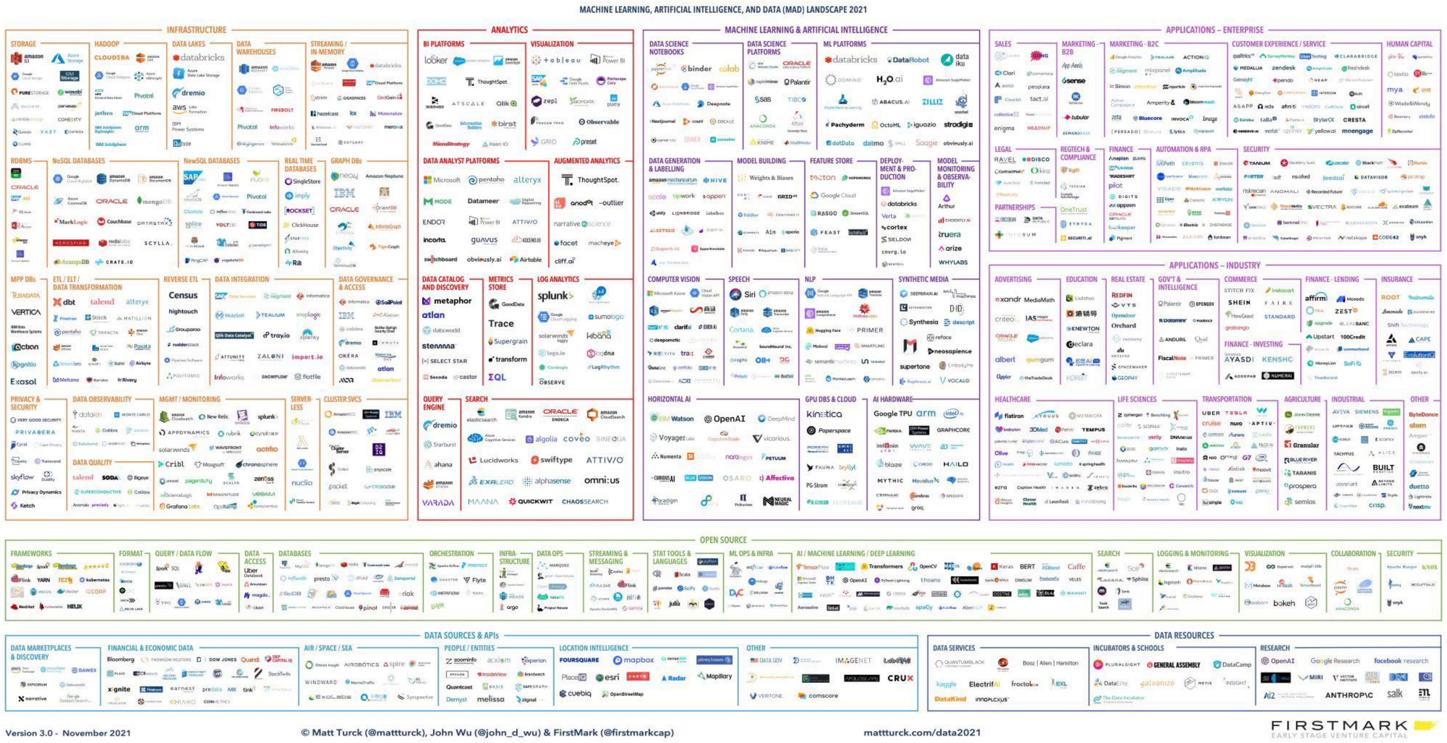
Source: https://www.mckinsey.com/business-functions/quantumblack/our-insights/the-data-driven-enterprise-of-2025

Big Data Landscape… Infrastructure is Changing

2012



2021


MACHINE LEARNING, ARTIFICIAL INTELLIGENCE, AND DATA (MAD) LANDSCAPE 2021
INFRASTRUCTURE
ANALYTICS
MACHINE LEARNING & ARTIFICIAL INTELLIGENCE
APPLICATIONS – ENTERPRISE
APPLICATIONS – INDUSTRY
OPEN SOURCE
DATA SOURCES & APIs
DATA RESOURCES
Version 3.0 - November 2021
© Matt Turck (@mattturck), John Wu (@john_d_wu) & FirstMark (@firstmarkcap)
mattturck.com/data2021
FIRSTMARK
EARLY STAGE VENTURE CAPITAL

New "technology", but same principles

Source: https://mattturck.com/data2021/

Some "birth-years". When was SQL born?

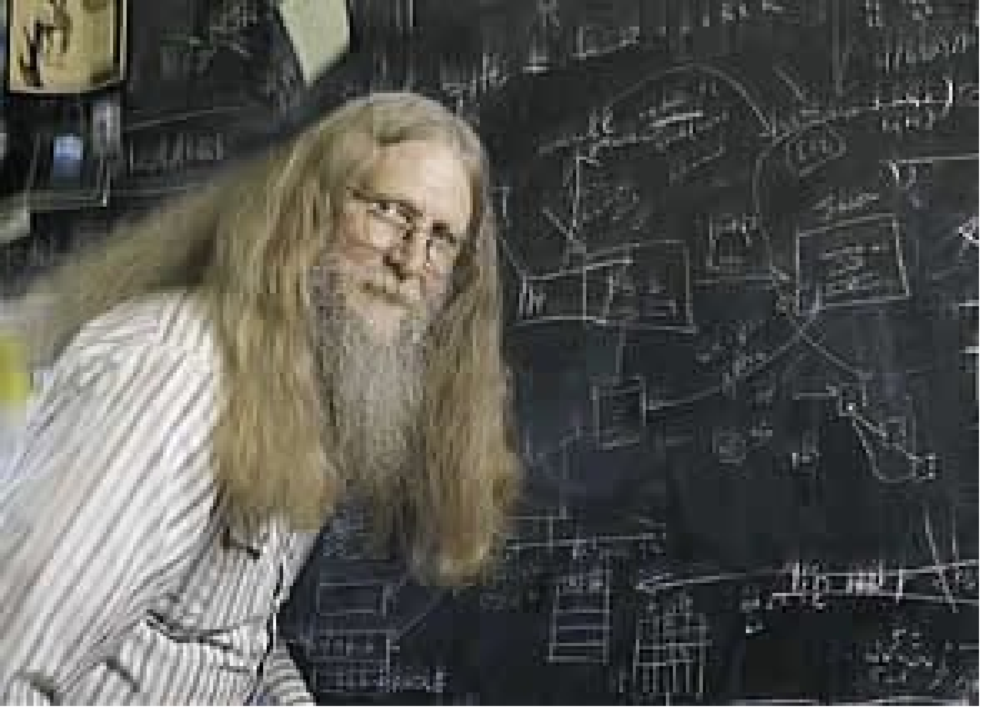
- 2006: Twitter
- 2004: Facebook
- 1998: Google
- 1995: Java, Ruby
- 1994: Amazon
- 1993: World Wide Web
- 1991: Python
- 1985: Windows

?

Some "birth-years"

- 2006: Twitter
- 2004: Facebook
- 1998: Google
- 1995: Java, Ruby
- 1994: Amazon
- 1993: World Wide Web
- 1991: Python
- 1985: Windows

- 1974: SQL

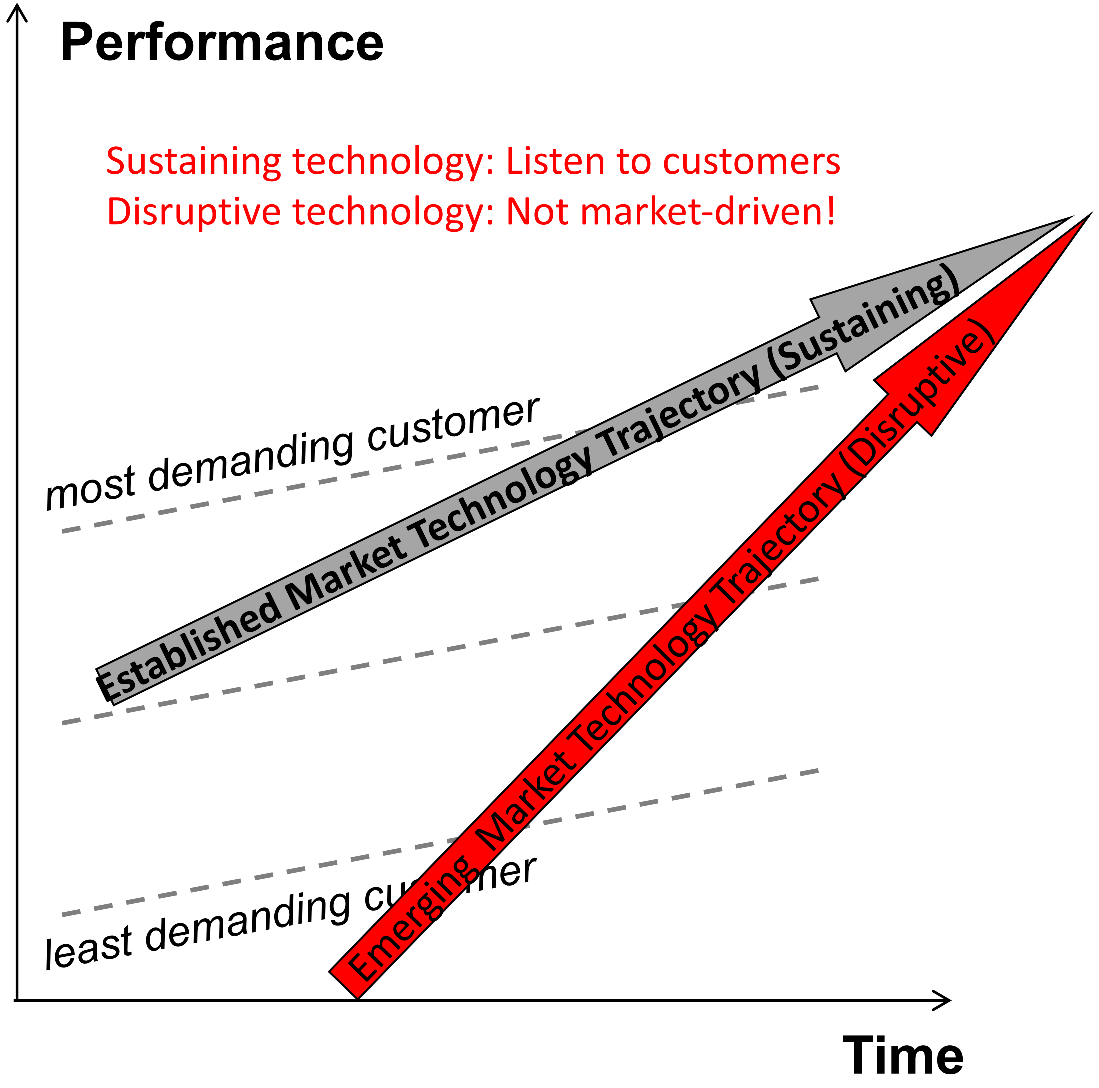


Bruce Lindsay, IBM Research

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

Source: As quoted in Winslett, Bruce Lindsay speaks out, SIGMOD record 2005. https://dl.acm.org/citation.cfm?id=1083803

Disruptive Innovation


Performance
Sustaining technology: Listen to customers
Disruptive technology: Not market-driven!
most demanding customer
Established Market Technology Trajectory (Sustaining)
Emerging Market Technology Trajectory (Disruptive)
least demanding customer
Time

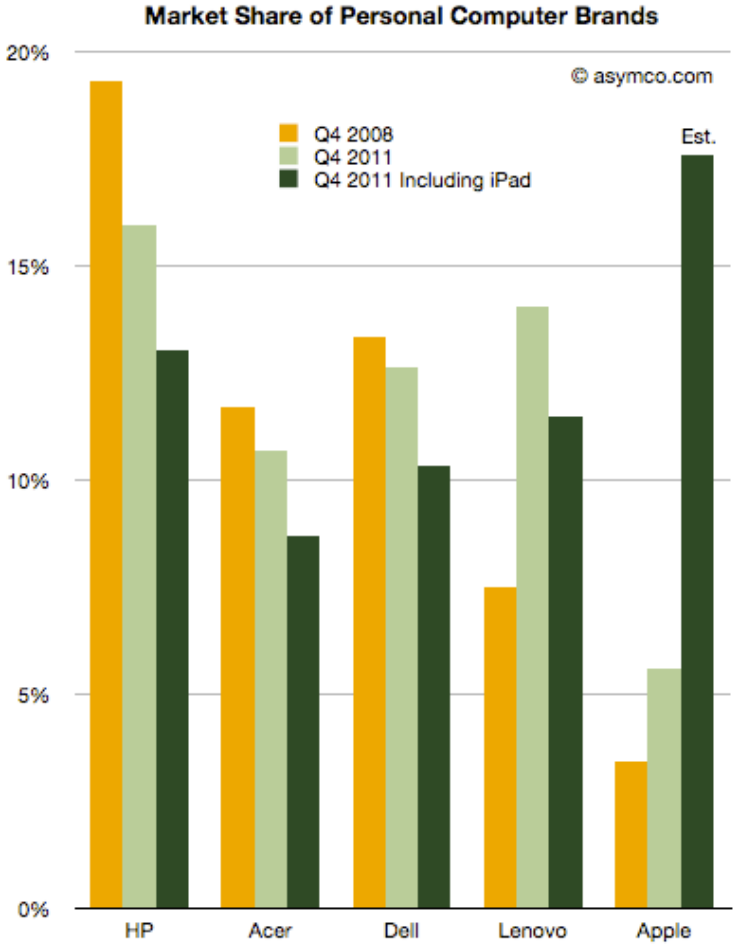
- Disruptive innovations are generally not acceptable for the mass market when they are introduced. Only the fringes of the market pick up the innovation in the first iteration
- It performs worse in one or more areas, but is typically simpler, more reliable, or more convenient than existing technologies.
- It is less profitable than existing technologies. Leading firms' most profitable customers generally can't use it and don't want it.
- As the innovator continues to refine their product the utility value to the market increases
- Its performance trajectory is steeper than that of existing technologies.
- Large organizations are fundamentally incapable of successfully bringing it to market.

Illustration and text compiled from “The Innovator’s Dilemma” by Clayton Christensen, 1997. https://en.wikipedia.org/wiki/The_Innovator%27s_Dilemma

iPhone: Disruptive Innovation or not?

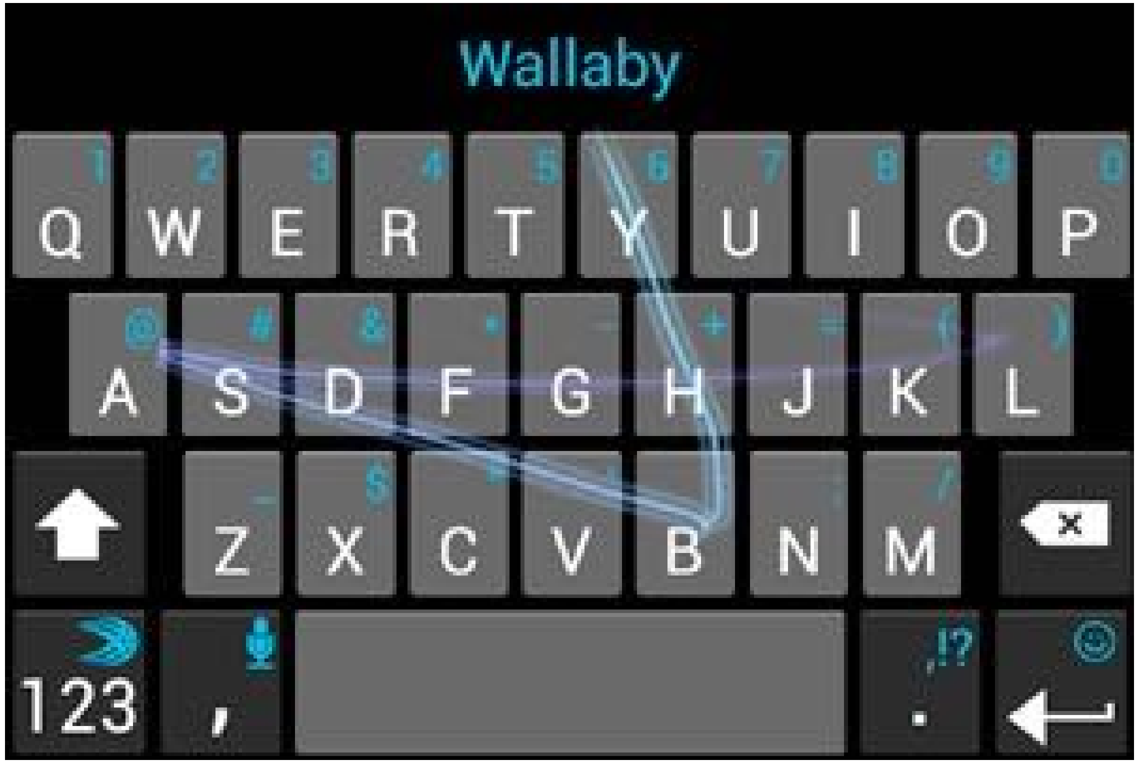
1: "Business Phones"
Microsoft in 2007

2: Laptops



Sources: http://www.youtube.com/watch?v=eywi0h_Y5_U, http://www.asymco.com/2012/01/16/apple-is-the-top-personal-computer-vendor/

What keyboards without keys can do...



In Feb 2016, SwiftKey was purchased by Microsoft, for 250 M$

swype

SwiftKey

Sources: https://en.wikipedia.org/wiki/Swype, https://en.wikipedia.org/wiki/SwiftKey

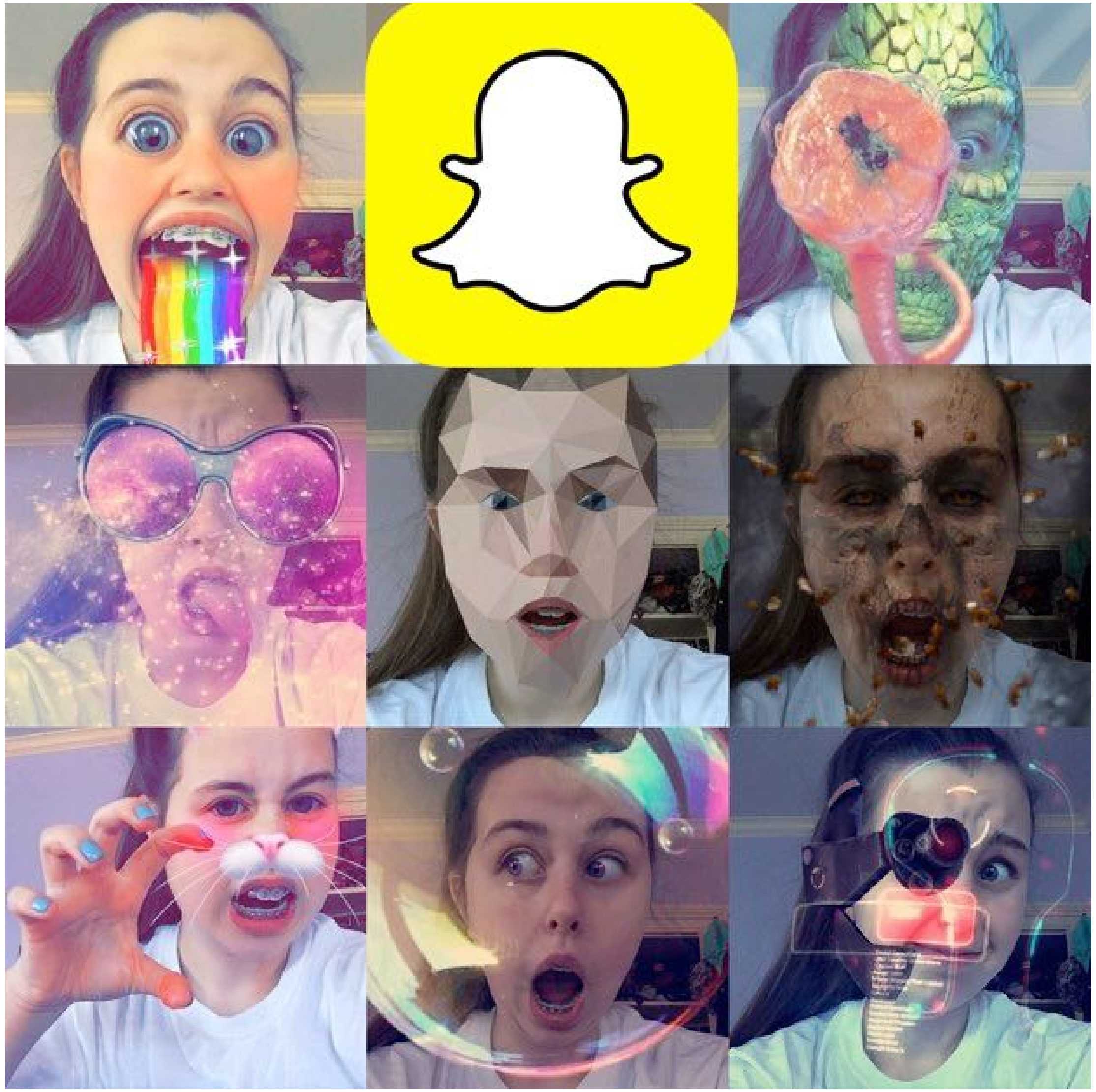
The keyboard of the future?

Source: http://www.wired.com/2014/06/siri_ai/

What can I help you with?
Messages
Photo

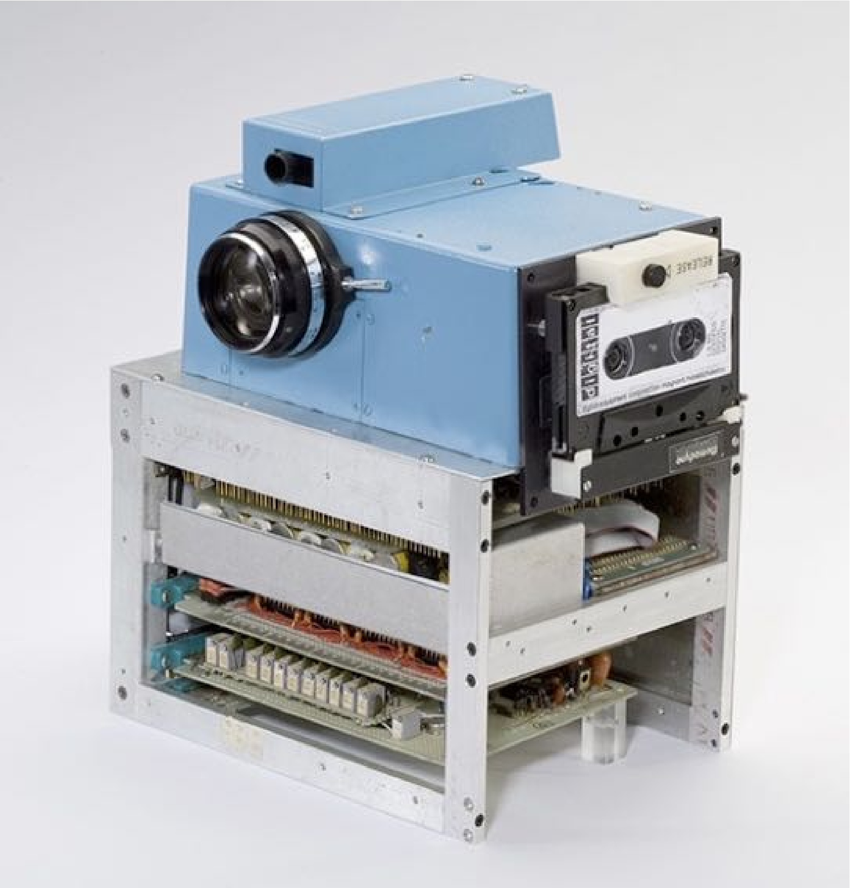
Eyes Free
MODE
engadget

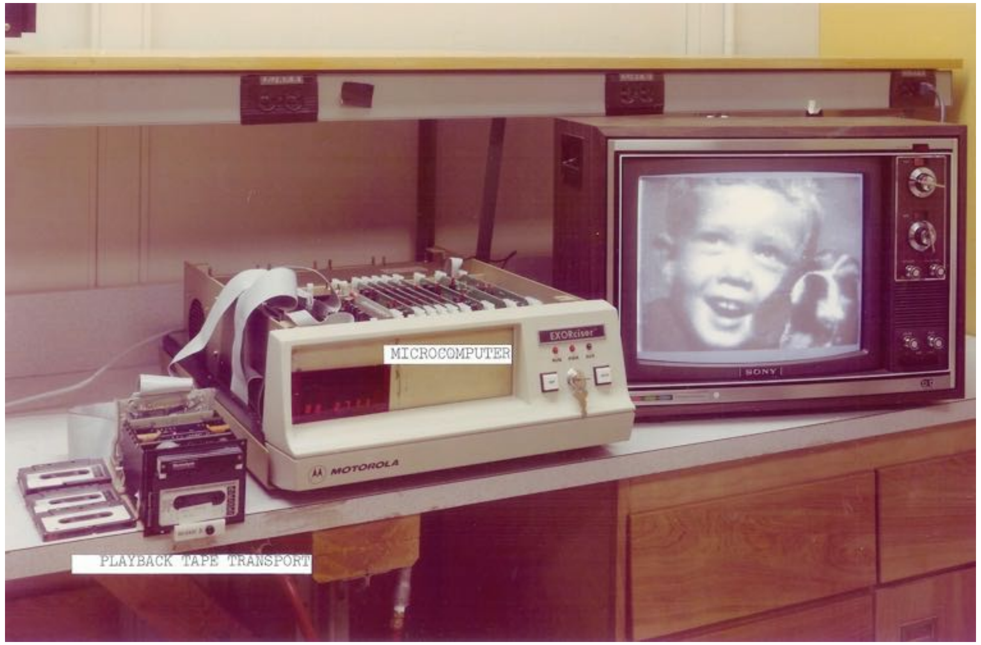
Keyboards? Do we need text to communicate?



Source: http://www.buzzfeed.com/benrosen/how-to-snapchat-like-the-teens (2/2016)

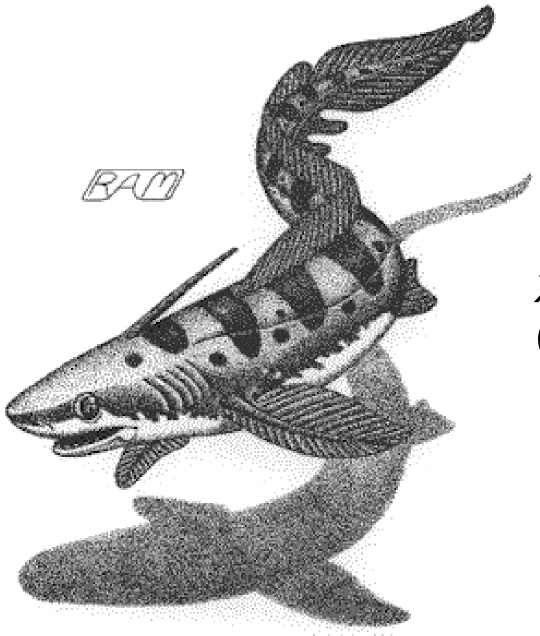
What is this? (1975)



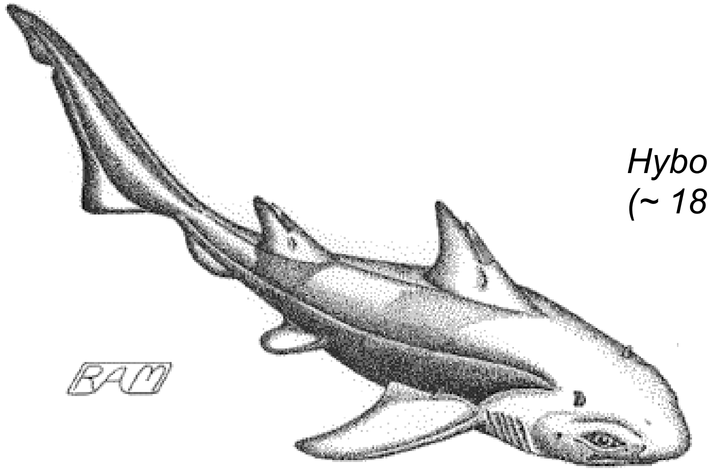


Source: http://pluggedin.kodak.com/pluggedin/post/?id=687843

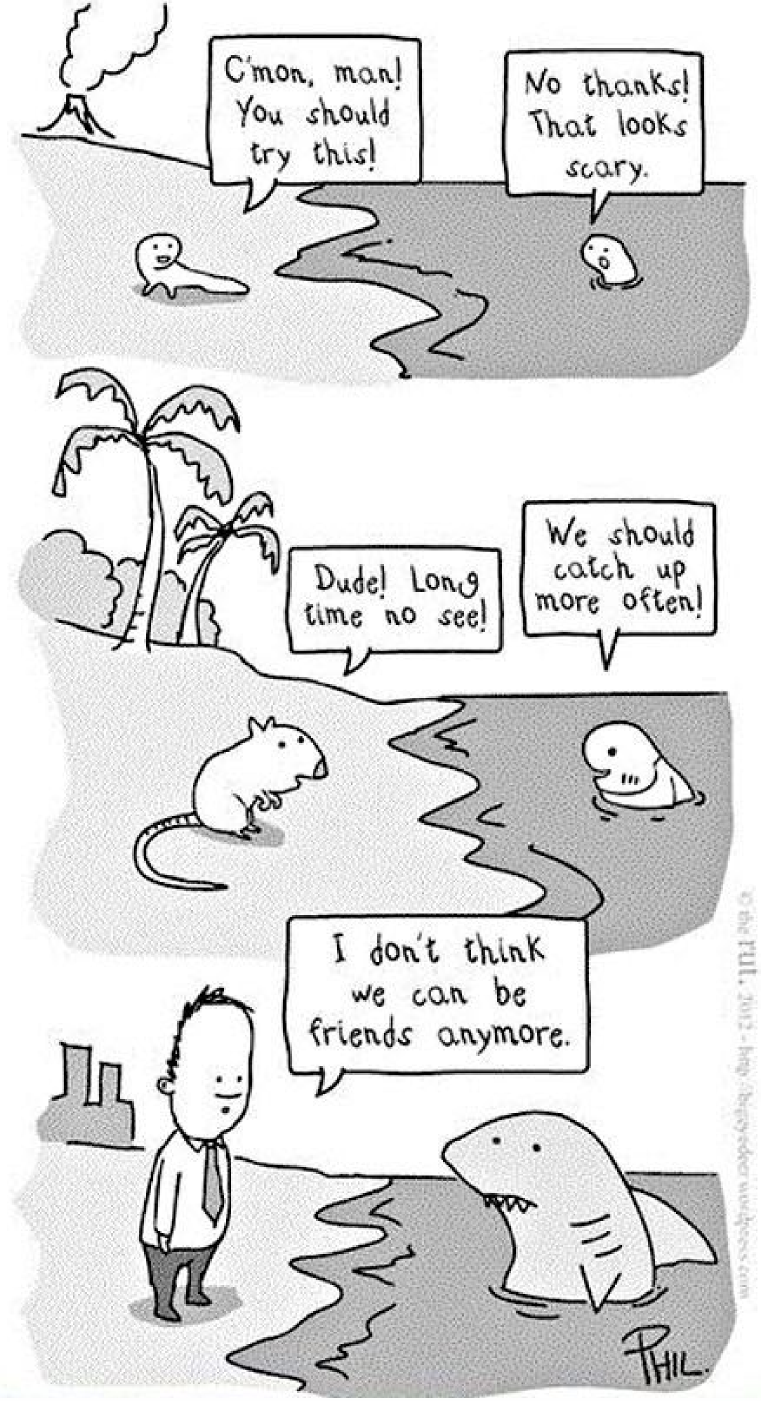
Evolution of Sharks



Xenacanthus
(~ 280 million years ago)

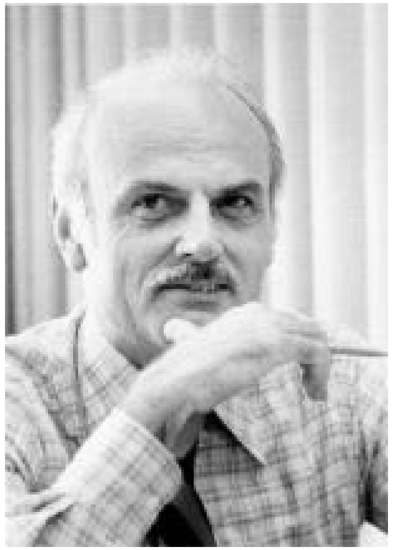


Hybodus sp.
(~ 180 million years ago)


C'mon, man! You should try this!
No thanks! That looks scary.
Dude! Long time no see!
We should catch up more often!
I don't think we can be friends anymore.
PHIL.

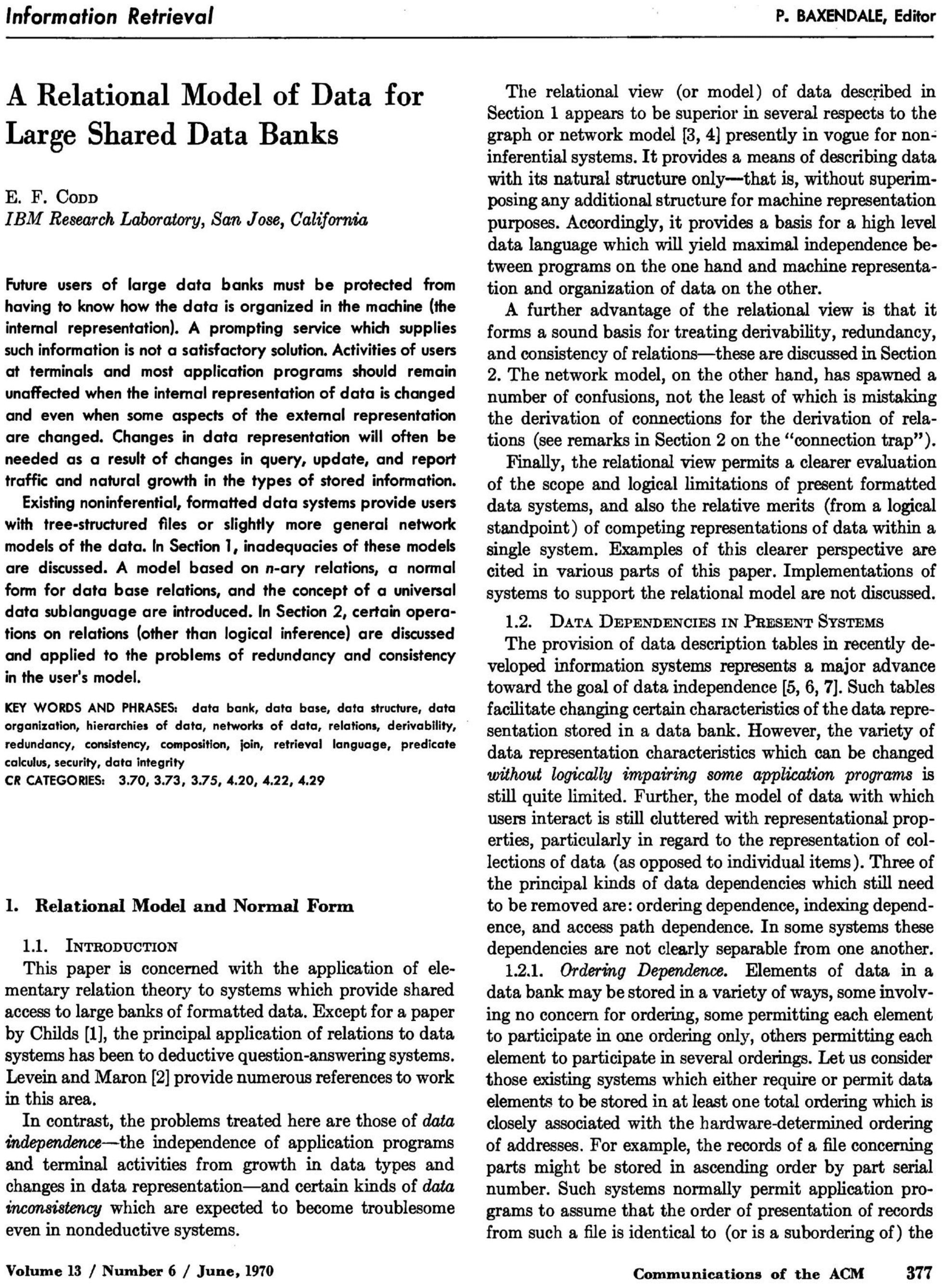
Source for picture on the right: http://themetapicture.com/evolution-it-makes-you-lose-friends/

SQL: some history



- Dr. Edgar Codd (IBM)
  - CACM June 1970: "A Relational Model of Data for Large Shared Data Banks" https://doi.org/10.1145/362384.362685
- Standardized
  - 1986 by ANSI: SQL1
  - 1992: Revised: SQL2
    - Approx 580 page document describing syntax and semantics
  - Revised: 1999, 2003, 2008, ...
- Players
  - Oracle (Relational Software), Microsoft, IBM, ...
- Every vendor has a slightly different version of SQL
- But the main commands are standardized

Codd's (disruptive ?) innovation



Information Retrieval P. BAXENDALE, Editor

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on *n*-ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

KEY WORDS AND PHRASES: data bank, data base, data structure, data organization, hierarchies of data, networks of data, relations, derivability, redundancy, consistency, composition, join, retrieval language, predicate calculus, security, data integrity
CR CATEGORIES: 3.70, 3.73, 3.75, 4.20, 4.22, 4.29

1. Relational Model and Normal Form

1.1. Introduction

This paper is concerned with the application of elementary relation theory to systems which provide shared access to large banks of formatted data. Except for a paper by Childs [1], the principal application of relations to data systems has been to deductive question-answering systems. Levein and Maron [2] provide numerous references to work in this area.

In contrast, the problems treated here are those of *data independence*—the independence of application programs and terminal activities from growth in data types and changes in data representation—and certain kinds of *data inconsistency* which are expected to become troublesome even in nondeductive systems.

The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for noninferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the "connection trap").

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of systems to support the relational model are not discussed.

1.2. Data Dependencies in Present Systems

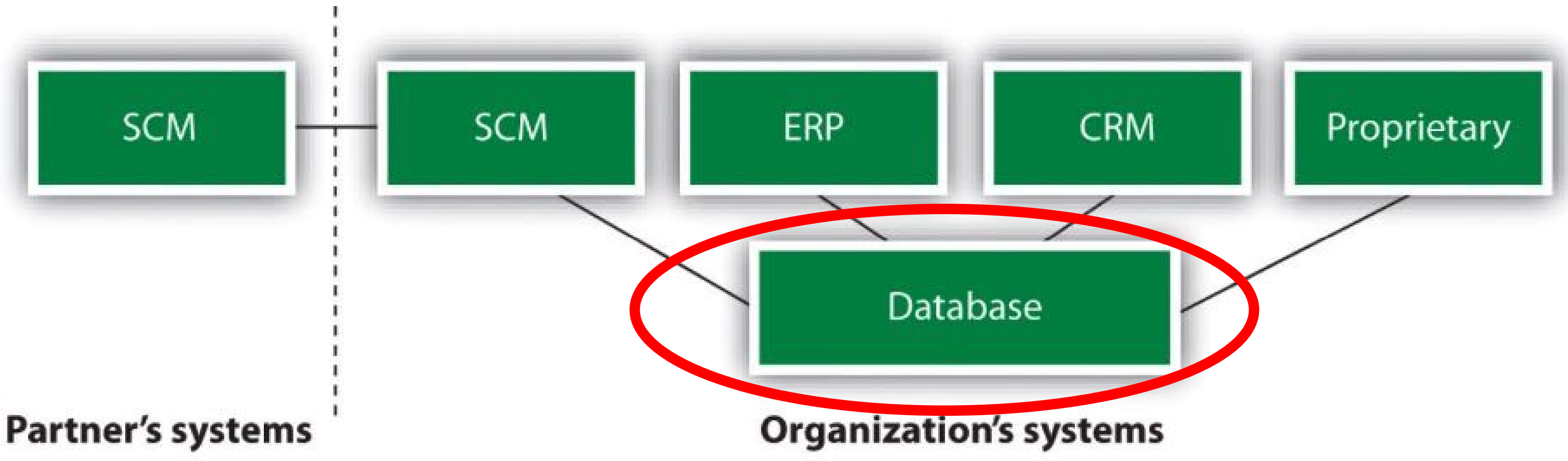
The provision of data description tables in recently developed information systems represents a major advance toward the goal of data independence [5, 6, 7]. Such tables facilitate changing certain characteristics of the data representation stored in a data bank. However, the variety of data representation characteristics which can be changed *without logically impairing some application programs* is still quite limited. Further, the model of data with which users interact is still cluttered with representational properties, particularly in regard to the representation of collections of data (as opposed to individual items). Three of the principal kinds of data dependencies which still need to be removed are: ordering dependence, indexing dependence, and access path dependence. In some systems these dependencies are not clearly separable from one another.

1.2.1. *Ordering Dependence.* Elements of data in a data bank may be stored in a variety of ways, some involving no concern for ordering, some permitting each element to participate in one ordering only, others permitting each element to participate in several orderings. Let us consider those existing systems which either require or permit data elements to be stored in at least one total ordering which is closely associated with the hardware-determined ordering of addresses. For example, the records of a file concerning parts might be stored in ascending order by part serial number. Such systems normally permit application programs to assume that the order of presentation of records from such a file is identical to (or is a subordering of) the

Volume 13 / Number 6 / June, 1970 Communications of the ACM 377

Source: Edgar F Codd. "A Relational Model of Data for Large Shared Data Banks ", CACM 1970. https://doi.org/10.1145/362384.362685

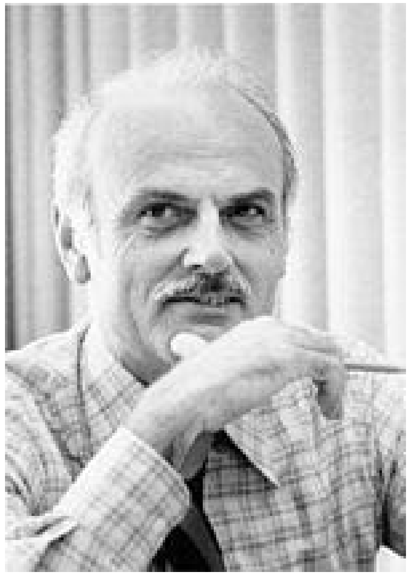
SQL and the relational model as standard



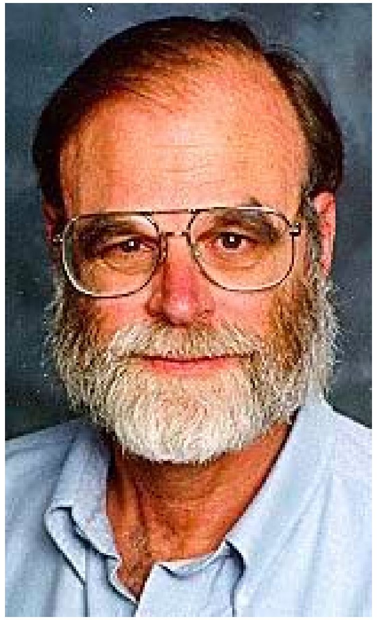
Source: "Information Systems: A Manager's Guide to Harnessing Technology (book v1.4)," p.185, Gallaugher, 2012. https://gallaugher.com/book/
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Five Turing Award Winners

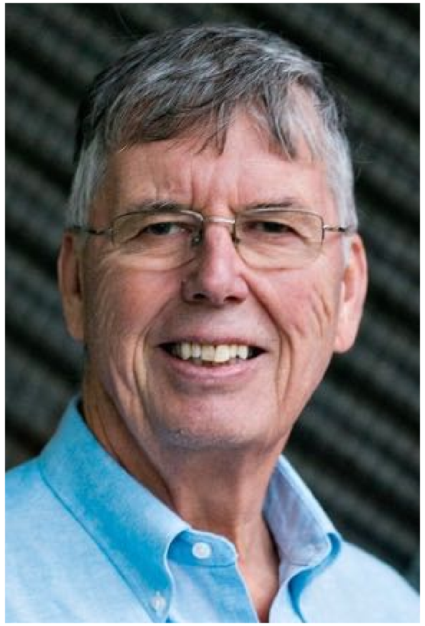
Charles
Bachmann
1973



Edgar
Codd
1981



Jim
Gray
1998



Michael
Stonebraker
2014

Jeffrey
Ullman
2020

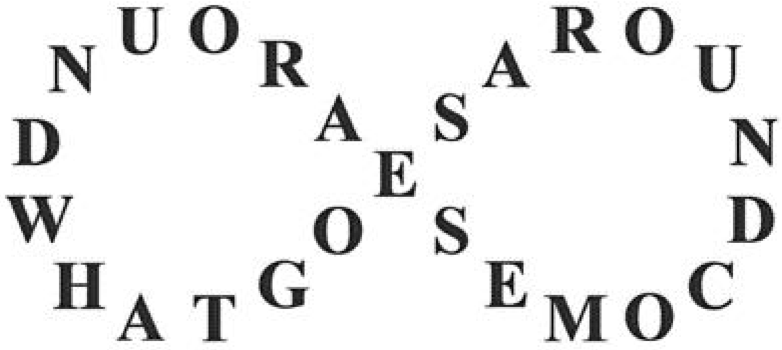
Compilers/PL/DB

Why should you study databases?

- Mercenary- make more $$$:
  - Startups need DB talent right away = low employee #
  - Massive industry…

Microsoft ORACLE IBM Google

- Intellectual:
  - Science: data poor to data rich
    - No idea how to handle the data!
  - Fundamental ideas to/from all of CS:
    - Systems, theory, AI, logic, stats, analysis….



Many great computer systems ideas started in DB.

Four topics we will cover

1. SQL: Relational data & Queries (~ 9 lectures)

- How can we query and manipulate data with SQL, a declarative language?
- Behind the curtain: by compiling high-level declarative queries into efficient low-level plans
  - reduced expressive power but the system can do more for you

2. Database Design: Design theory and constraints (~ 7 lectures)

- How can we collect and **organize** large amounts of data?
- By designing relational schemas that can efficiently index and organize your data, and keeps the data from getting corrupted

Lectures: from a user's perspective & how it works

3. Transactions: Syntax & supporting systems (~ 3 lectures)

- How can we manage **concurrent access** to data as it is read and written?
- A programmer's abstraction for data consistency

4. Other Data models & query processing/optimization (~ 3 lectures)

- How do databases actually optimize queries? What are other **data models**?
- Relational algebra, Basics of query optimization (Cost Estimates)
- External Memory Algorithms (IO model) for sorting and joins
- "NoSQL": Key-Value Stores, Column Stores, Document stores, Graph DBs
- TBD: Pandas, data modeling in Excel

What this course is about (and what not)

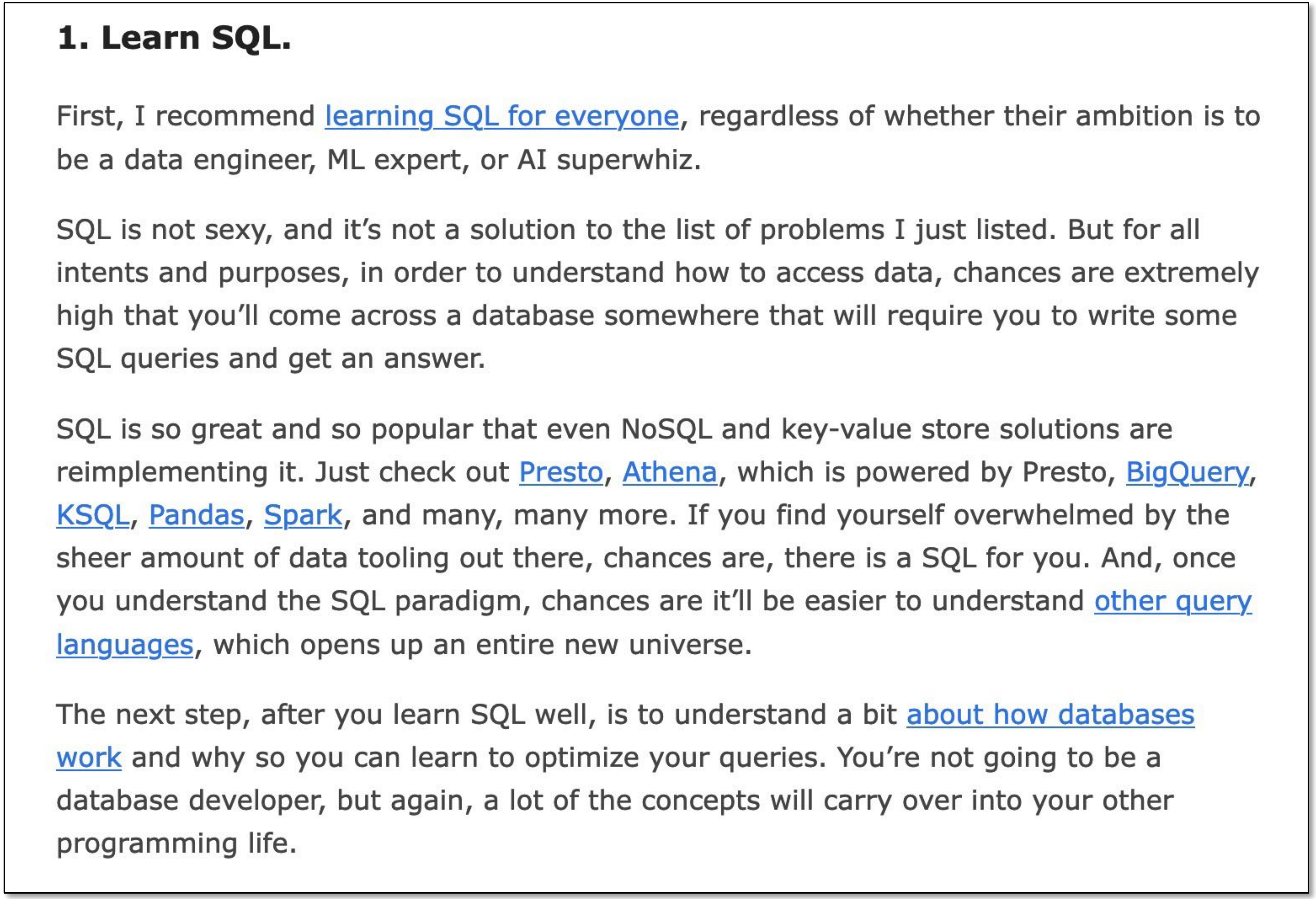
- Discuss **fundamentals of data management**
  - How to design databases (2) and query databases (1)
  - Not how to be a Database Administrator (DBA), or how to tune Oracle 12g, or to build a website with MySQL

- We cover some of **how database management systems work** and the principles of scalable data management
  - Inconsistency in distributed settings (3)
  - A little bit of what goes on behind the curtain (4). The background is too diverse to go into depth
  - See cs4200 Database internals, taught by Renee Miller, for a more advanced version for cs3200

When you'll use this material

- Building almost any software application
  - e.g., mobile, cloud, consumer, enterprise, analytics, machine learning
  - Corollary: every application you use uses a database
  - Bonus: every program consumes data (even if only the program text!)
- Performing data analytics
  - Business intelligence, data science, predictive modeling
  - (Even if you're using Pandas https://pandas.pydata.org/, you're using relational algebra!)

- Building data-intensive tools and applications
  - Many core concepts power deep learning frameworks to self-driving cars



1. Learn SQL.

First, I recommend learning SQL for everyone, regardless of whether their ambition is to be a data engineer, ML expert, or AI superwhiz.

SQL is not sexy, and it's not a solution to the list of problems I just listed. But for all intents and purposes, in order to understand how to access data, chances are extremely high that you'll come across a database somewhere that will require you to write some SQL queries and get an answer.

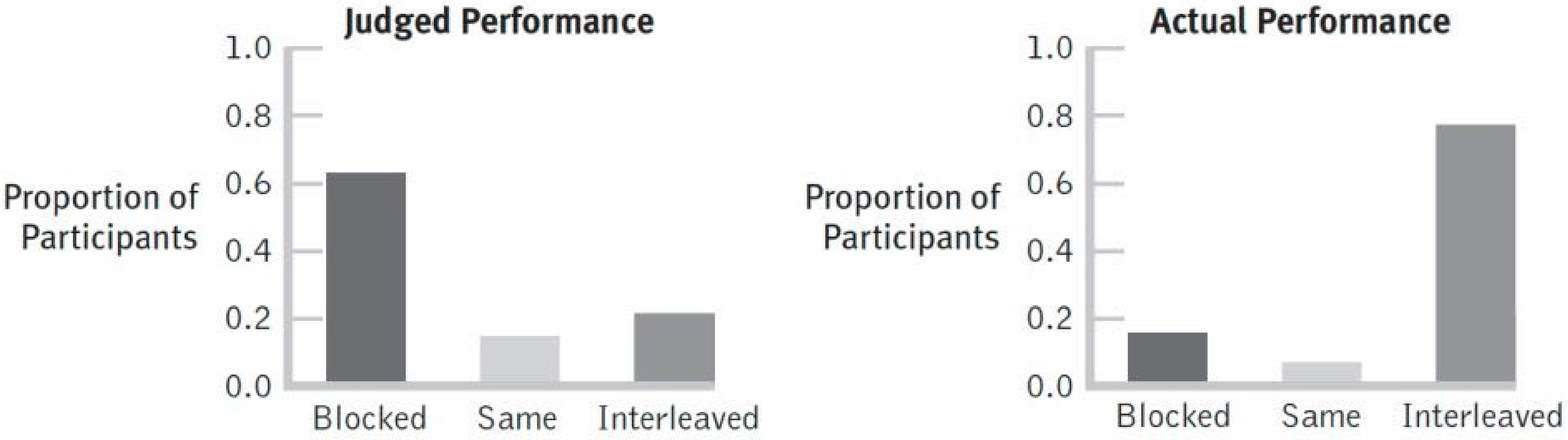
SQL is so great and so popular that even NoSQL and key-value store solutions are reimplementing it. Just check out Presto, Athena, which is powered by Presto, BigQuery, KSQL, Pandas, Spark, and many, many more. If you find yourself overwhelmed by the sheer amount of data tooling out there, chances are, there is a SQL for you. And, once you understand the SQL paradigm, chances are it'll be easier to understand other query languages, which opens up an entire new universe.

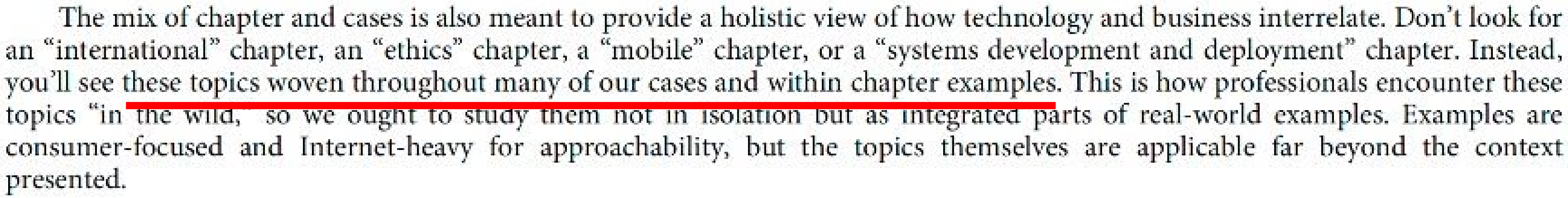
The next step, after you learn SQL well, is to understand a bit about how databases work and why so you can learn to optimize your queries. You're not going to be a database developer, but again, a lot of the concepts will carry over into your other programming life.

Source: https://vickiboykis.com/2019/02/13/data-science-is-different-now/

Course Pedagogy

Sequencing Material: "Under which teaching condition do you think you learn better?"

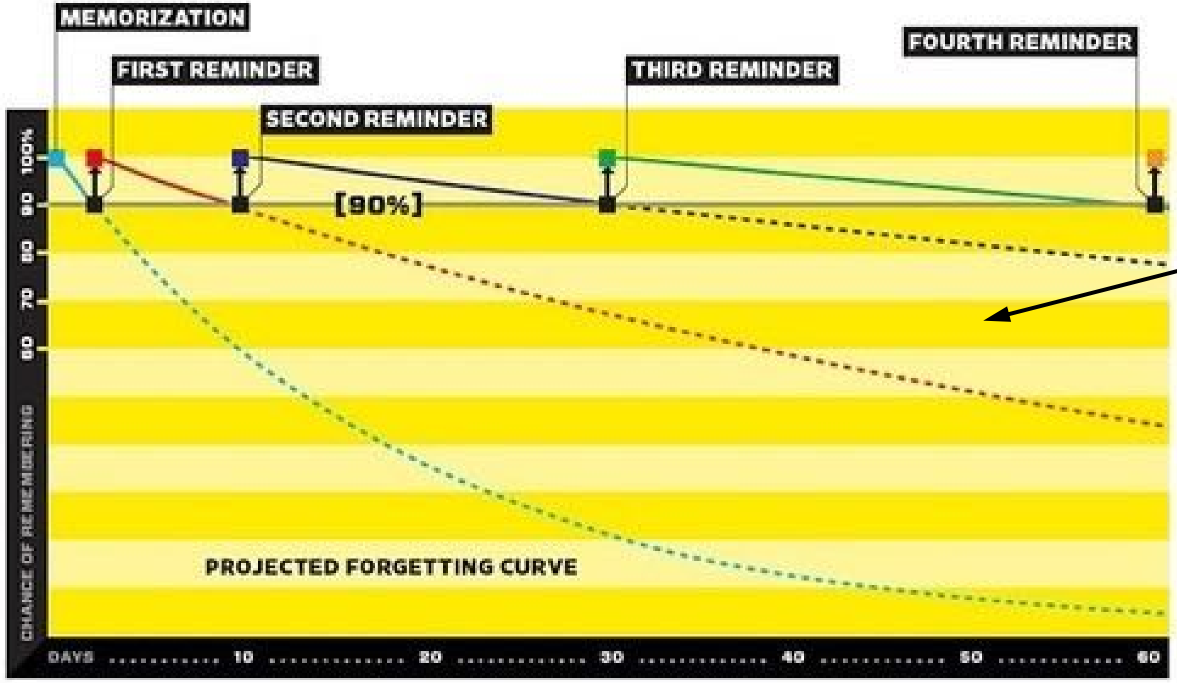




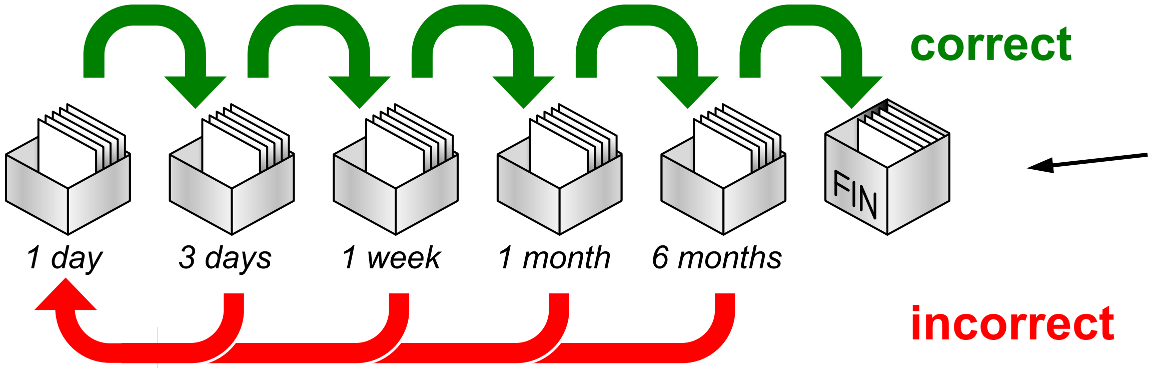
The mix of chapter and cases is also meant to provide a holistic view of how technology and business interrelate. Don't look for an "international" chapter, an "ethics" chapter, a "mobile" chapter, or a "systems development and deployment" chapter. Instead, you'll see these topics woven throughout many of our cases and within chapter examples. This is how professionals encounter these topics "in the wild," so we ought to study them not in isolation but as integrated parts of real-world examples. Examples are consumer-focused and Internet-heavy for approachability, but the topics themselves are applicable far beyond the context presented.

Data from: Bjork & Bjork, "Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning," 2011. https://psycnet.apa.org/record/2011-19926-008
Paragraph from:"Information Systems: A Manager's Guide to Harnessing Technology (book v1.4)," Gallaugher, 2012. https://gallaugher.com/book/

Spaced Repetition



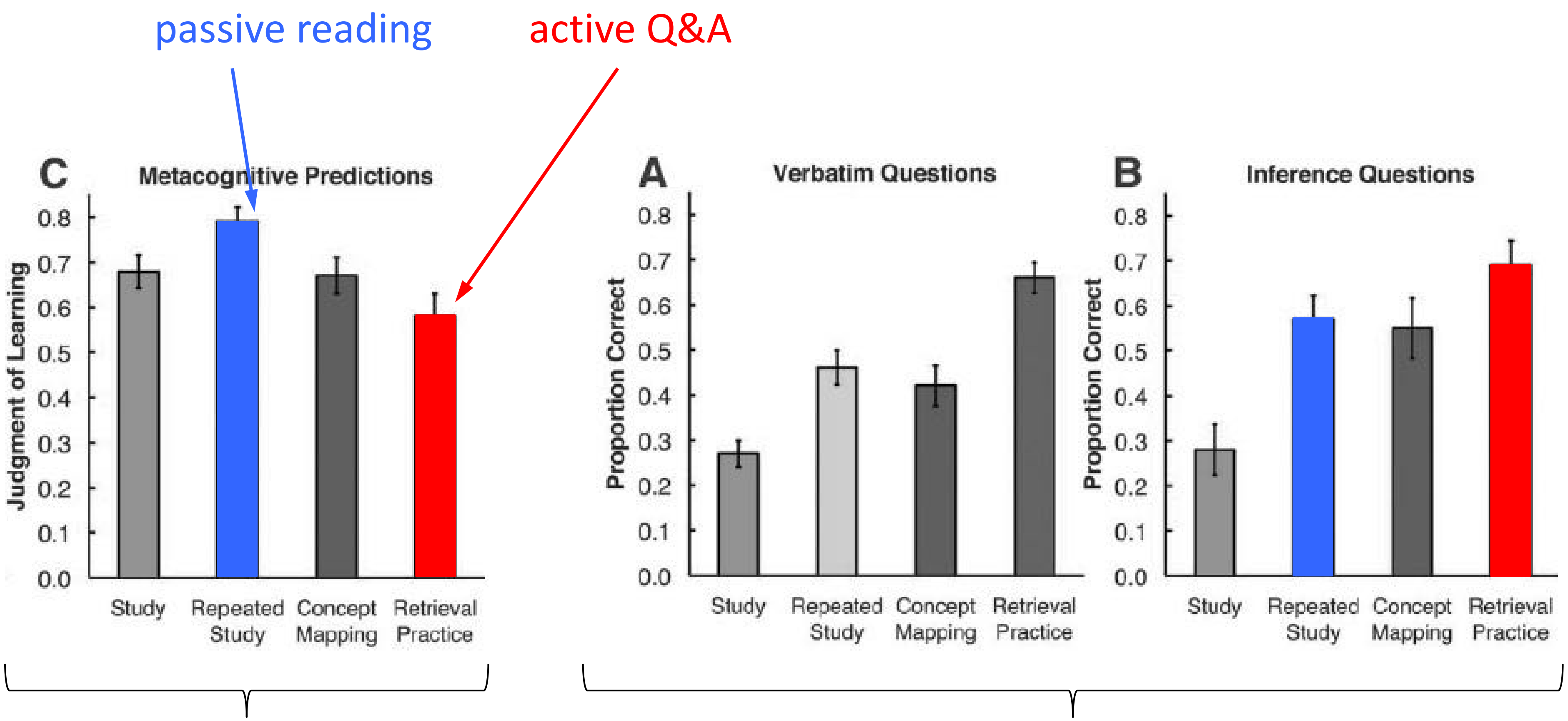
Ebbinghaus Forgetting Curve



Leitner System (Pimsleur's graduated interval recall)

Sources: http://www.wired.com/2008/04/ff-wozniak/, Gatterbauer & Suciu, "Managing Structured Collections of Community Data," CIDR 2011.

Studying new material: "Under which study condition do you think you learn better?"



passive reading

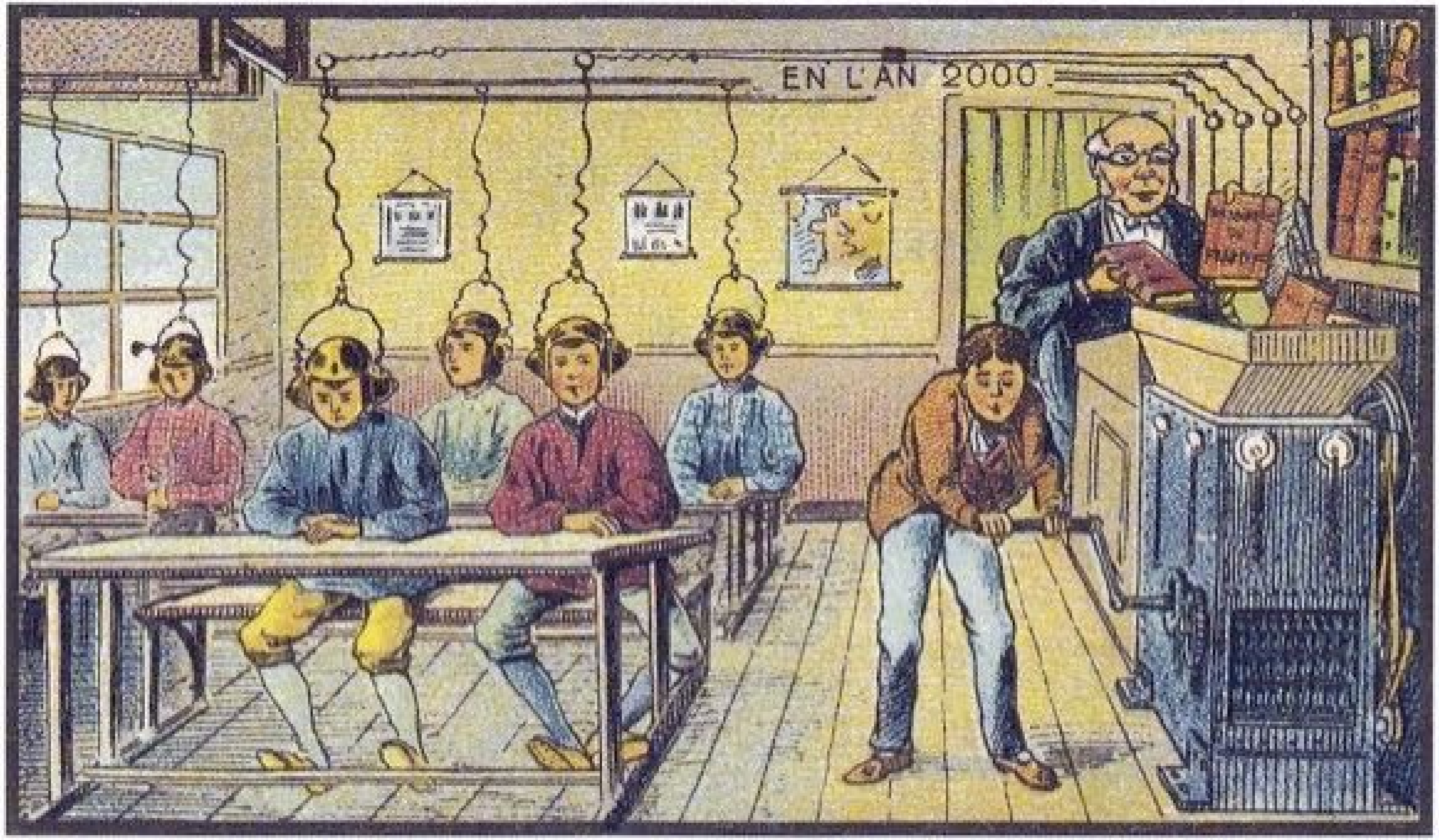
active Q&A

Judged performance
(=what people think)

Actual performance
(=what is actually working)

Data from: Karpicke & Blunt, "Retrieval Practice Produces More Learning than Elaborative Studying with Concept Mapping," Science, 2011. https://doi.org/10.1126/science.1199327

The year 2000 imagined in 1900

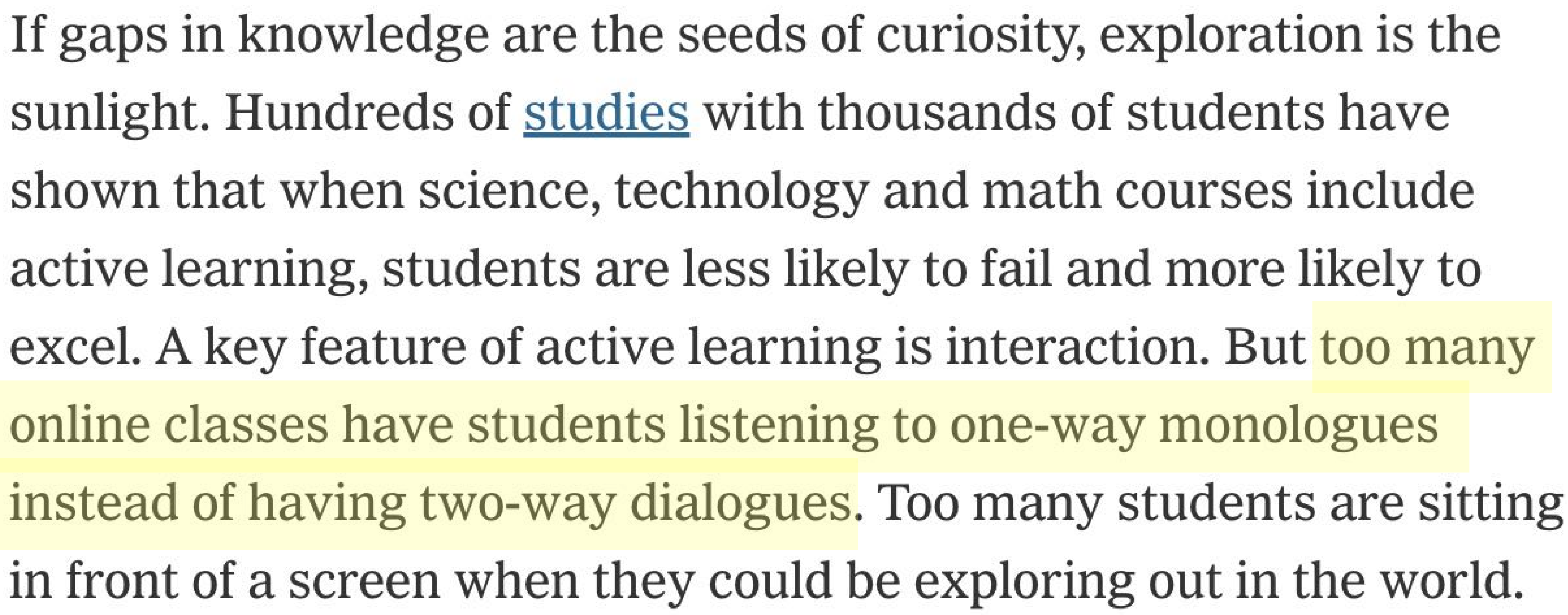


At School

Source: https://publicdomainreview.org/collection/a-19th-century-vision-of-the-year-2000

Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Lectures are not recorded (1/2)



If gaps in knowledge are the seeds of curiosity, exploration is the sunlight. Hundreds of studies with thousands of students have shown that when science, technology and math courses include active learning, students are less likely to fail and more likely to excel. A key feature of active learning is interaction. But too many online classes have students listening to one-way monologues instead of having two-way dialogues. Too many students are sitting in front of a screen when they could be exploring out in the world.

Source: https://www.nytimes.com/2020/09/07/opinion/remote-school.html

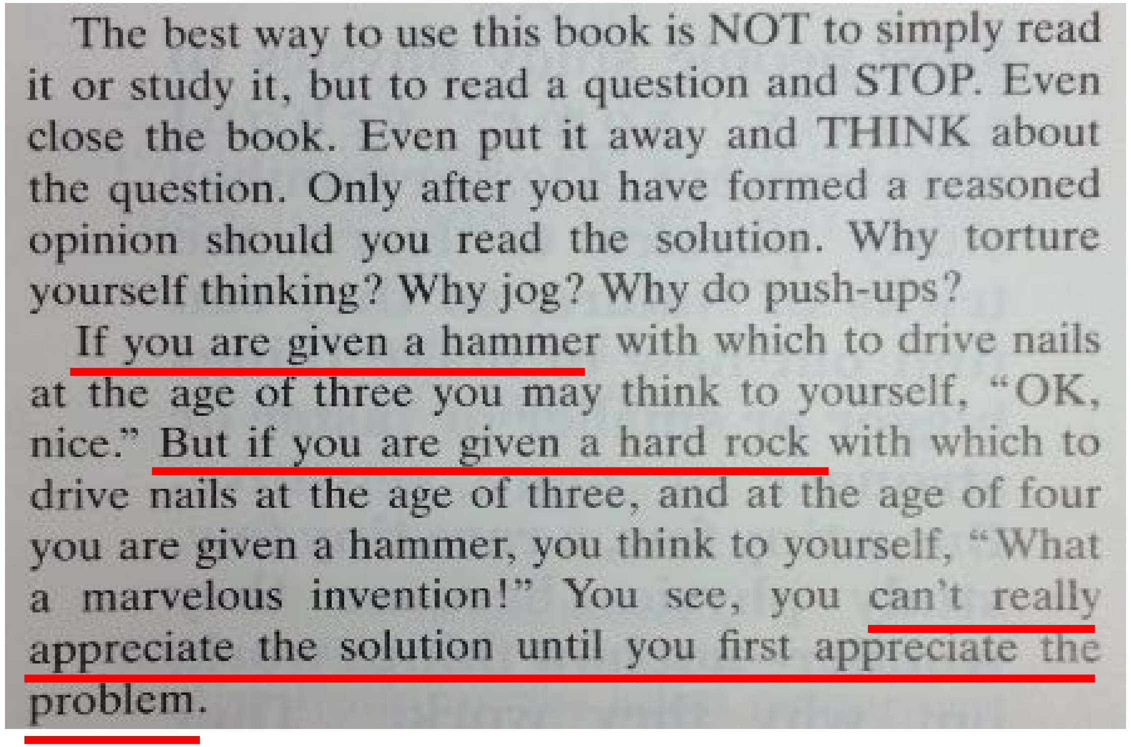
Lectures are not recorded (2/2)

- We would like to have an encouraging environment in which everyone can speak up and discuss ideas freely without concern that discussions will be available outside of classroom.
- The course slides are comprehensive and should allow you to be able to remember the key lessons from class (except for background stories I may tell you). Lecture slides will be posted after each class, usually by end of next day.
- Do not record or otherwise share the classroom video calls yourself. The Commonwealth of Massachusetts's wiretapping law requires "two-party consent". It is a felony to secretly record a conversation, whether the conversation is in person or taking place by telephone or another electronic medium. [See Mass. Gen. Laws ch.272, § 99].

Mass. Gen. Laws ch. 272, § 99: https://malegislature.gov/laws/generallaws/partiv/titlei/chapter272/section99

One reason why I don't post slides *before* lecture

From the preamble of one of the best physics books ever: „How to read this book"

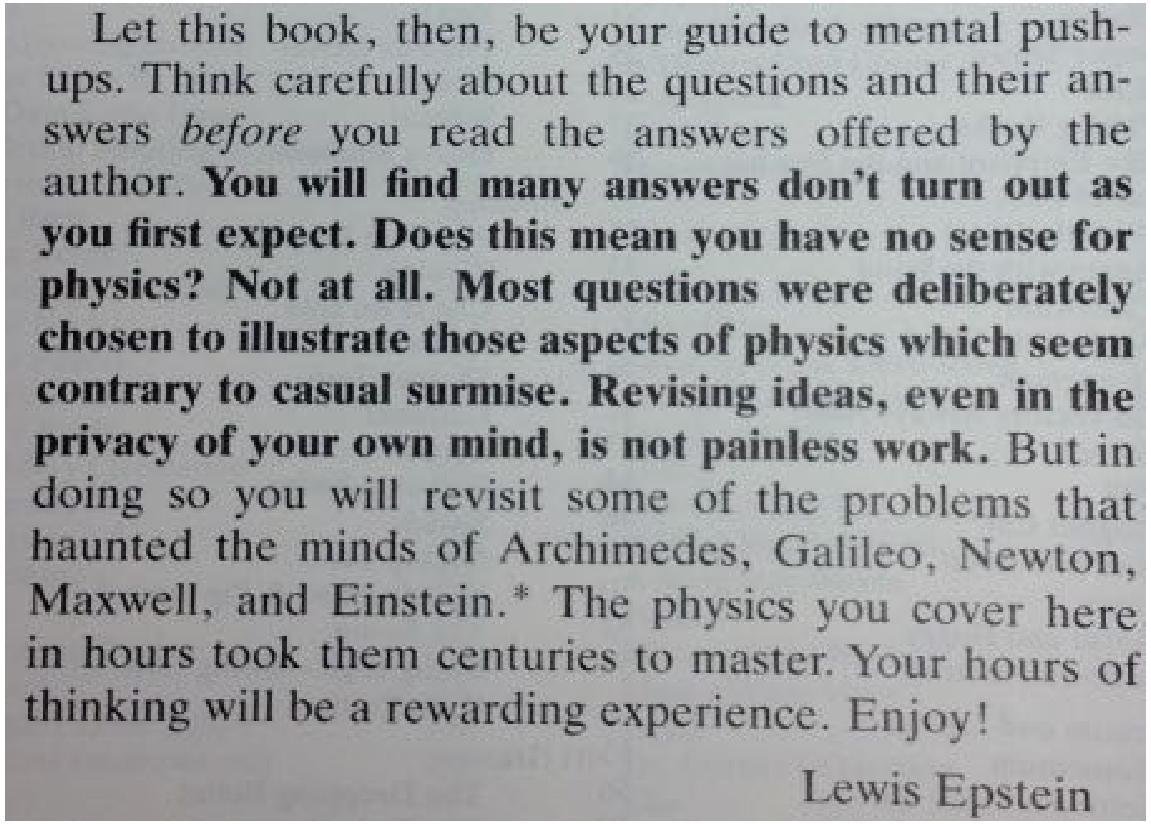


The best way to use this book is NOT to simply read it or study it, but to read a question and STOP. Even close the book. Even put it away and THINK about the question. Only after you have formed a reasoned opinion should you read the solution. Why torture yourself thinking? Why jog? Why do push-ups?

If you are given a hammer with which to drive nails at the age of three you may think to yourself, "OK, nice." But if you are given a hard rock with which to drive nails at the age of three, and at the age of four you are given a hammer, you think to yourself, "What a marvelous invention!" You see, you can't really appreciate the solution until you first appreciate the problem.

...

...



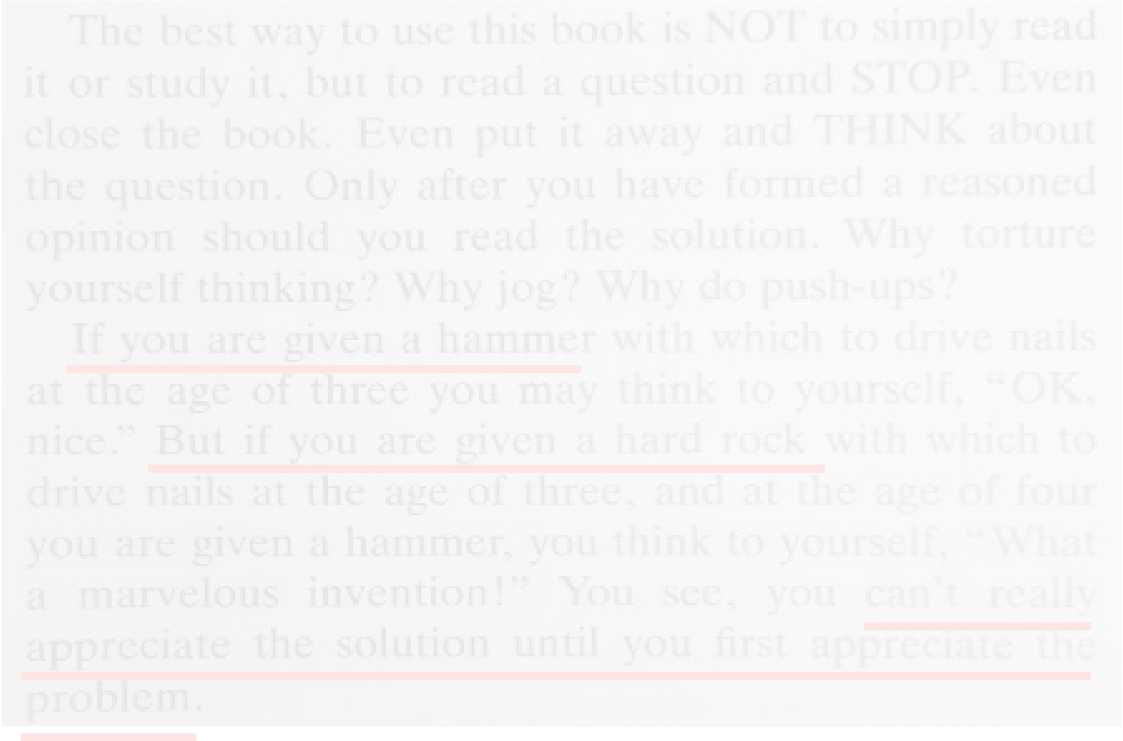
Let this book, then, be your guide to mental push-ups. Think carefully about the questions and their answers *before* you read the answers offered by the author. **You will find many answers don't turn out as you first expect. Does this mean you have no sense for physics? Not at all. Most questions were deliberately chosen to illustrate those aspects of physics which seem contrary to casual surmise. Revising ideas, even in the privacy of your own mind, is not painless work.** But in doing so you will revisit some of the problems that haunted the minds of Archimedes, Galileo, Newton, Maxwell, and Einstein.* The physics you cover here in hours took them centuries to master. Your hours of thinking will be a rewarding experience. Enjoy!

Lewis Epstein

Source: "Thinking Physics: Understanding Practical Reality", Lewis Carroll Epstein, 1979-2009. https://www.goodreads.com/book/show/268266.Thinking_Physics

One reason why I don't post slides *before* lecture

From the preamble of one of the best physics books ever: „How to read this book"



The best way to use this book is NOT to simply read it or study it, but to read a question and STOP. Even close the book. Even put it away and THINK about the question. Only after you have formed a reasoned opinion should you read the solution. Why torture yourself thinking? Why jog? Why do push-ups?

If you are given a hammer with which to drive nails at the age of three you may think to yourself, "OK, nice." But if you are given a hard rock with which to drive nails at the age of three, and at the age of four you are given a hammer, you think to yourself, "What a marvelous invention!" You see, you can't really appreciate the solution until you first appreciate the problem.

…

…

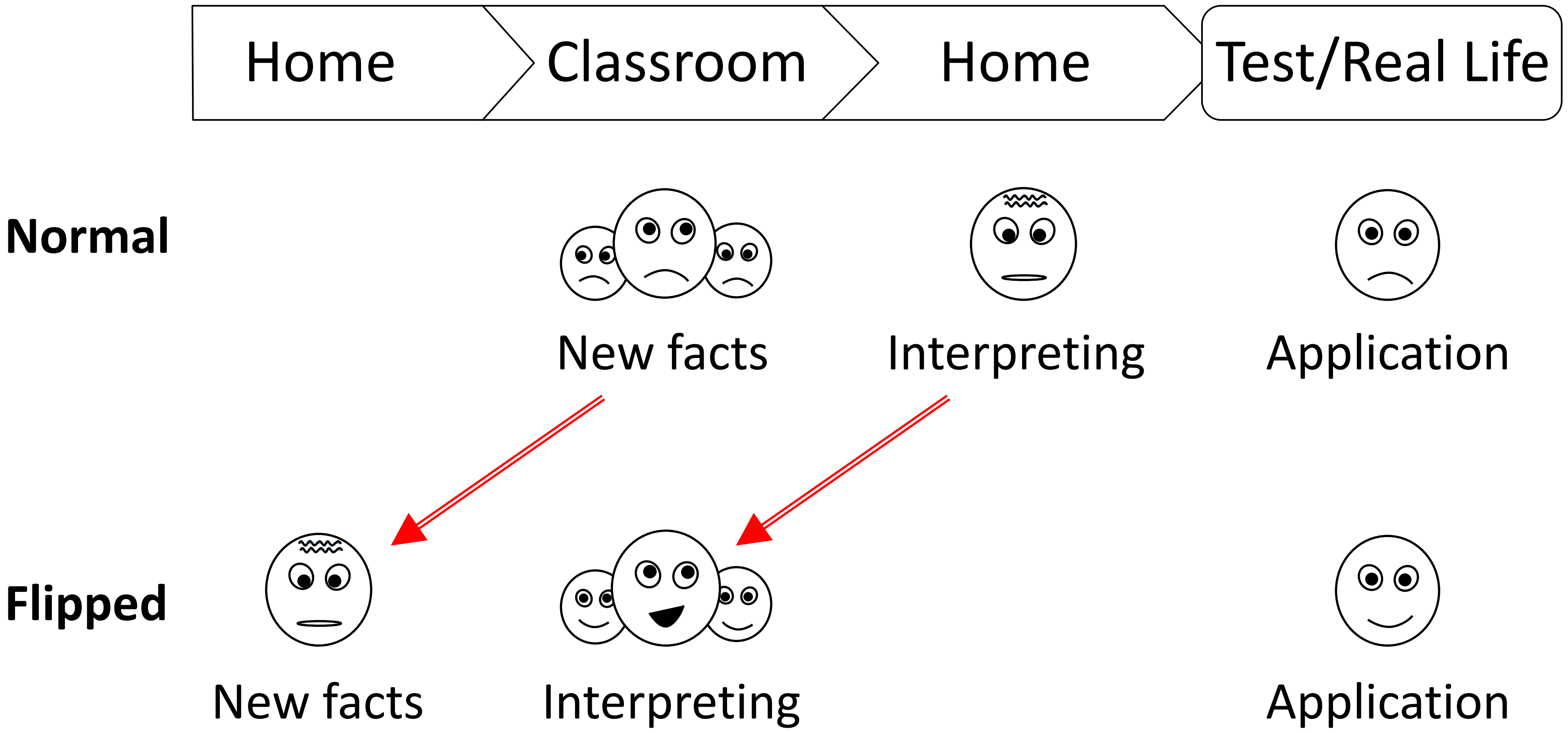
You must avoid the temptation to look at answers until you have tried to find and ideally write out the solution yourself!

Source: "Thinking Physics: Understanding Practical Reality", Lewis Carroll Epstein, 1979-2009. https://www.goodreads.com/book/show/268266.Thinking_Physics

The "Surfer Analogy" for time management

Source: http://stwww.surfermag.com/files/2013/10/Yak_Charlie-970x646.jpg

Former Classroom Philosophy



Home → Classroom → Home → Test/Real Life

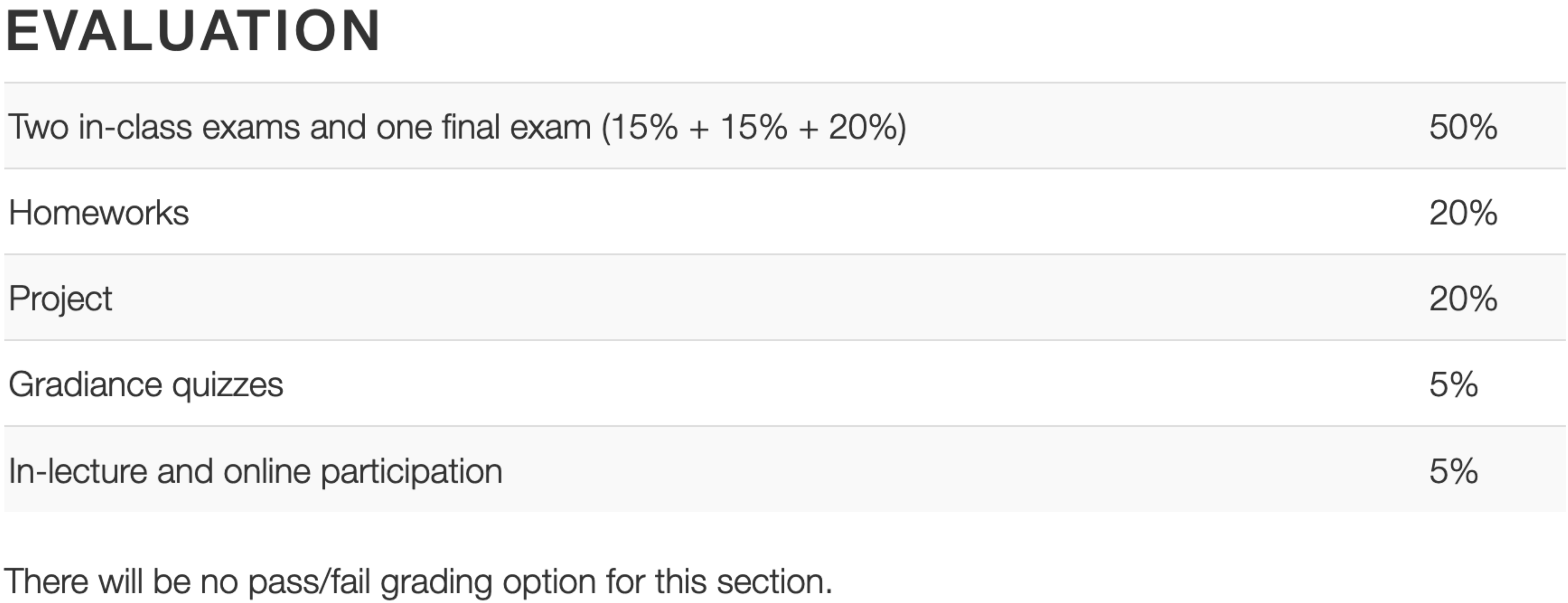
Normal: New facts, Interpreting, Application

Flipped: New facts, Interpreting, Application

"Flipped Classroom" but requires preparation before coming to class.

In my experience works better for graduate than undergraduate classes.

Source: [Gatterbauer, "Interpreting Instead of Teaching Facts in the Classroom," Teaching Theory and Academic Writing, 2008.]()



EVALUATION

Two in-class exams and one final exam (15% + 15% + 20%)	50%
Homeworks	20%
Project	20%
Gradiance quizzes	5%
In-lecture and online participation	5%

There will be no pass/fail grading option for this section.

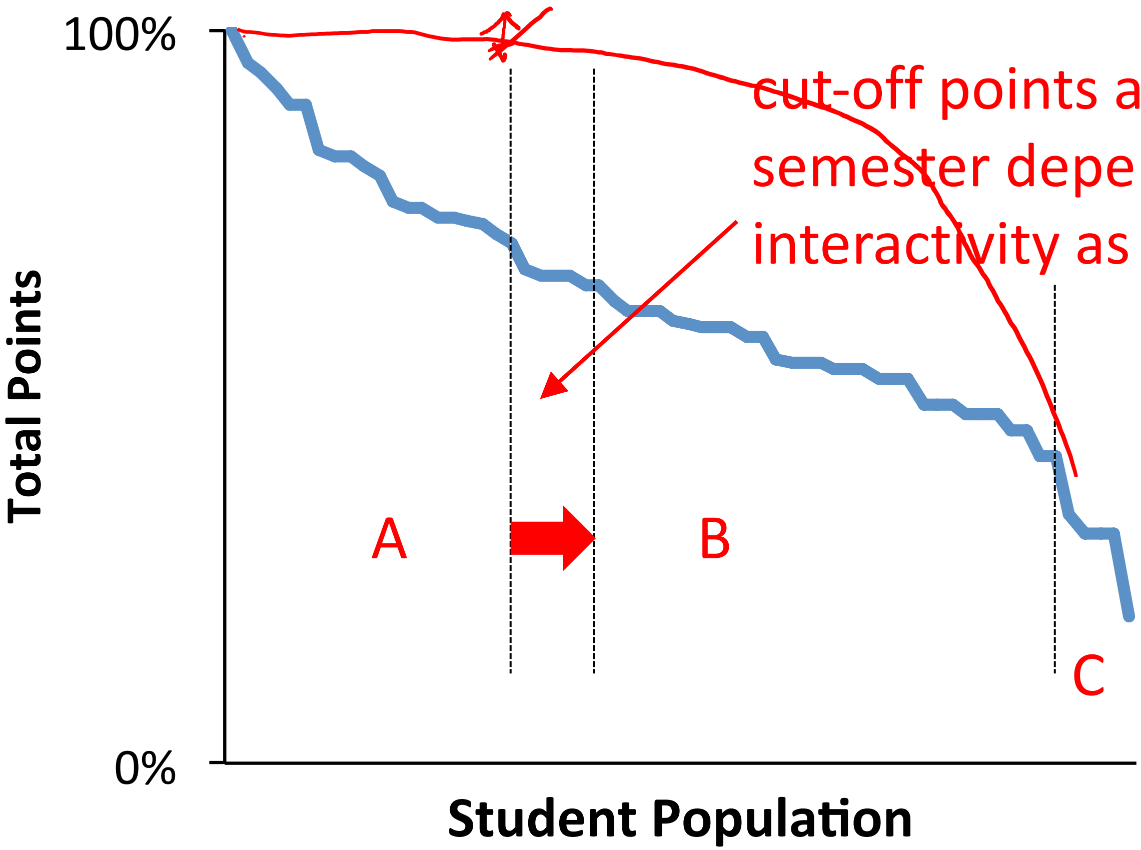
My pedagogic goals for classroom effectiveness

Goal	Increased learning	Fair assessment
Metric	$\nearrow \dfrac{\Delta \text{ learning}}{\text{time invested}}$ ratio	$\nearrow \dfrac{\text{signal}}{\text{noise}}$ ratio
Implications	minimize chores, have "group-enhanced" HWs, "soft-graded" HWs, no attendance check, in-class problems, class contributions, interleaved, discuss interesting student solutions, ...	exam: hard (by score), comprehensive, individual, time-constrained
Risks	"Slacking off"	Stress, "not fun"

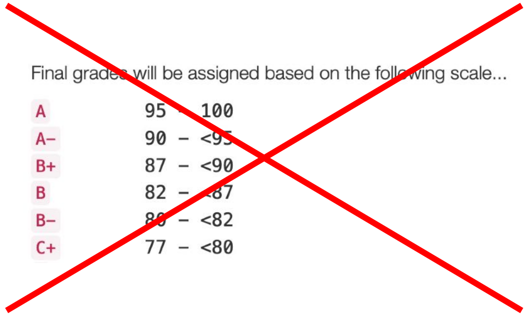
Grading Philosophy

Actual point distribution from a past final exam: long, but fair!

- no fixed percentages (e.g., top 30% get A)
- no fixed cut-offs (e.g., 80/100 points for A)



cut-off points at the end of the semester depend on overall class interactivity as compared to other years


Final grades will be assigned based on the following scale...

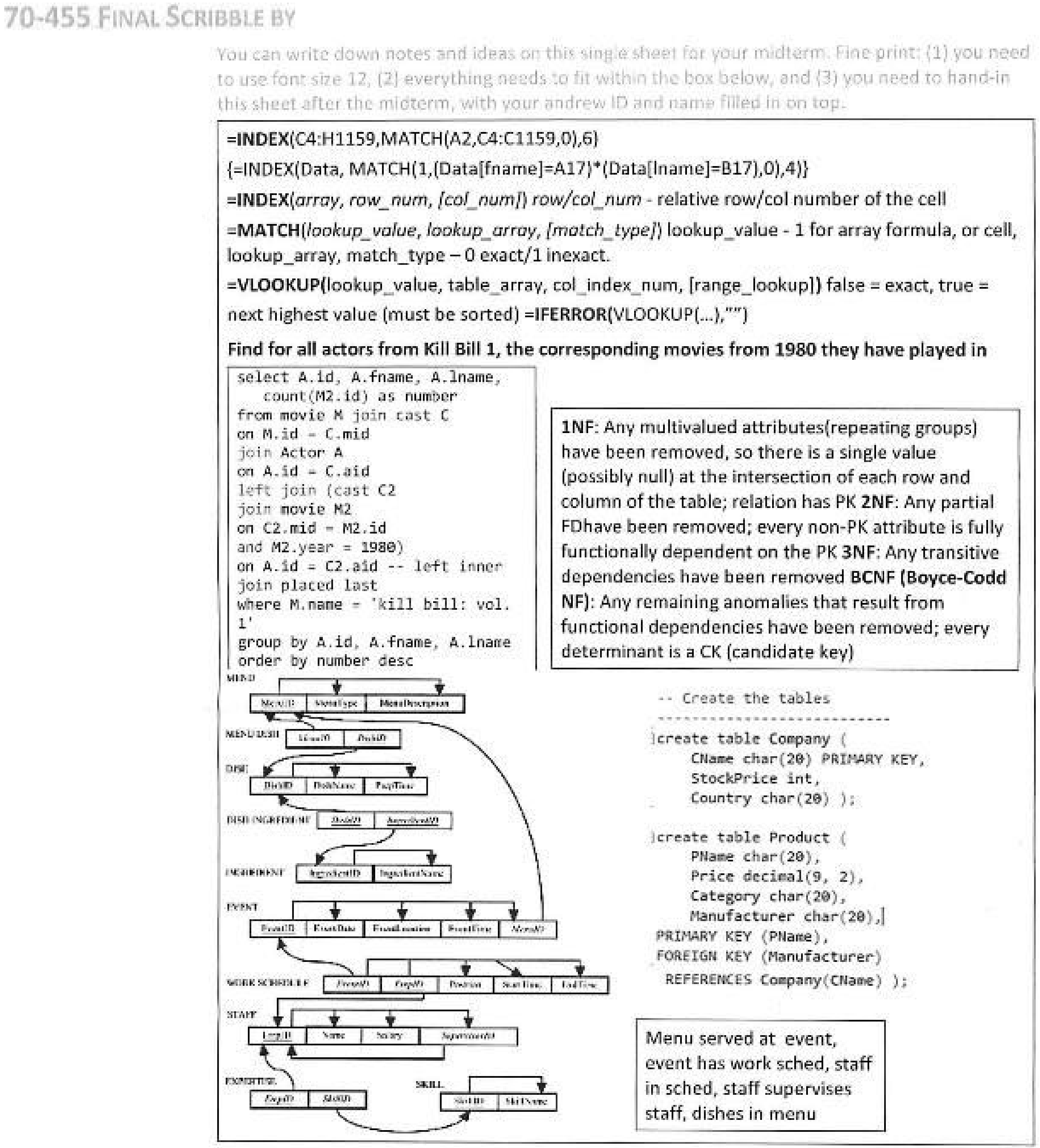
A	95 – 100
A–	90 – <95
B+	87 – <90
B	82 – <87
B–	80 – <82
C+	77 – <80

Contrast <u>external grades vs. internal scores</u>: Grade distributions are similar to other sections, but scores distributions are not

I will not disclose the actual cut-off points. Don't ask for an exception.

Final exam

- Open book but individual, hard, comprehensive, time-constrained
- Therefore: fair!
- See HW4 for past exam 2
- But don't mistake the dominant "group-enabled" work throughtout the term with you knowing the material alone
- As synthesis exercise, you need to make and bring a one-page cheat sheet and hand it in after the exam



70-455 FINAL SCRIBBLE BY

You can write down notes and ideas on this single sheet for your midterm. Fine print: (1) you need to use font size 12, (2) everything needs to fit within the box below, and (3) you need to hand-in this sheet after the midterm, with your andrew ID and name filled in on top.

=INDEX(C4:H1159,MATCH(A2,C4:C1159,0),6)

{=INDEX(Data, MATCH(1,(Data[fname]=A17)*(Data[lname]=B17),0),4)}

=INDEX(*array, row_num, [col_num]*) *row/col_num* - relative row/col number of the cell

=MATCH(*lookup_value, lookup_array, [match_type]*) lookup_value - 1 for array formula, or cell, lookup_array, match_type – 0 exact/1 inexact.

=VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup]) false = exact, true = next highest value (must be sorted) =**IFERROR**(VLOOKUP(...),"")

Find for all actors from Kill Bill 1, the corresponding movies from 1980 they have played in

```
select A.id, A.fname, A.lname,
   count(M2.id) as number
from movie M join cast C
on M.id = C.mid
join Actor A
on A.id = C.aid
left join (cast C2
join movie M2
on C2.mid = M2.id
and M2.year = 1980)
on A.id = C2.aid -- left inner
join placed last
where M.name = 'kill bill: vol.
1'
group by A.id, A.fname, A.lname
order by number desc
```

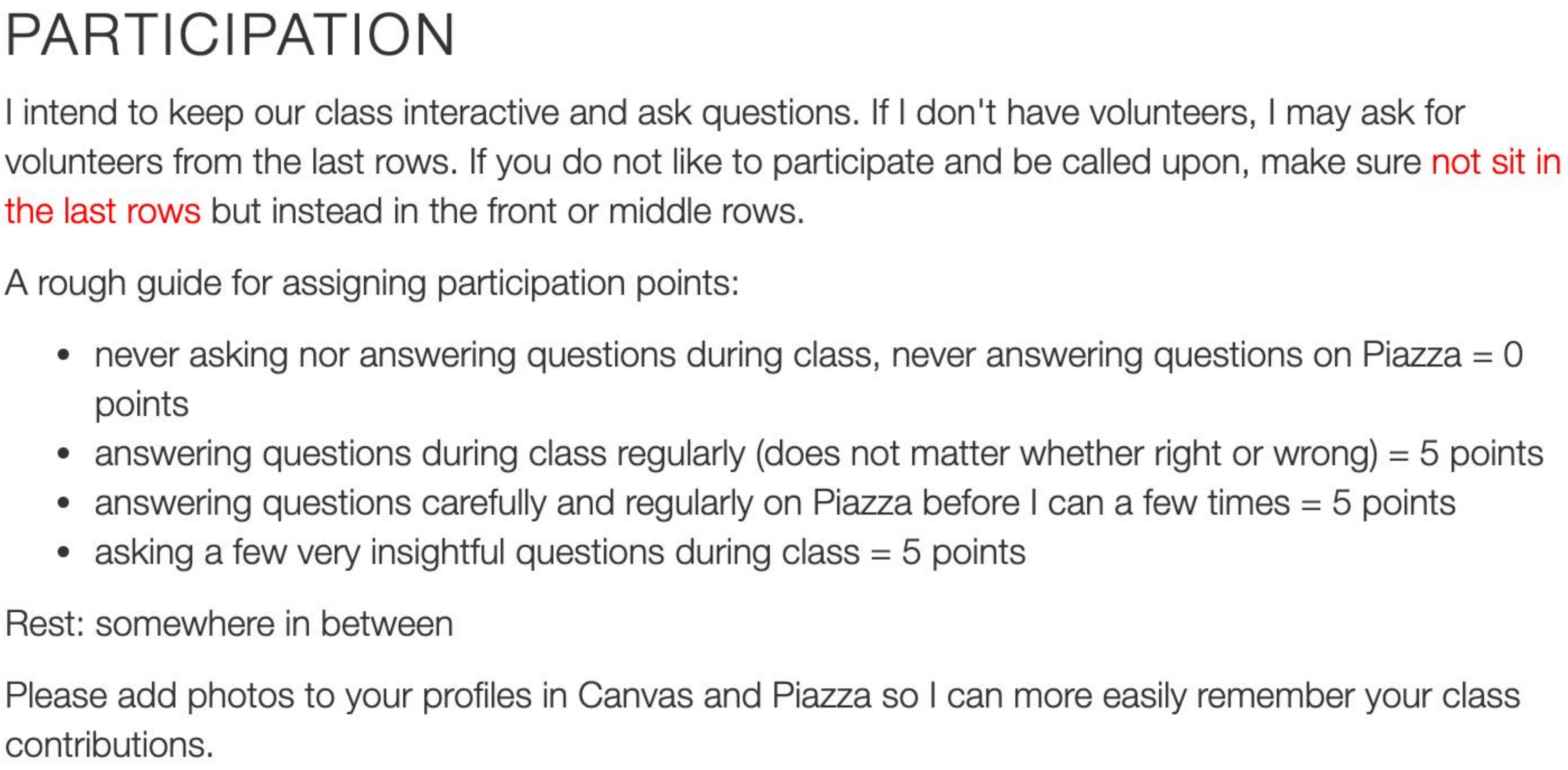
1NF: Any multivalued attributes(repeating groups) have been removed, so there is a single value (possibly null) at the intersection of each row and column of the table; relation has PK **2NF**: Any partial FDhave been removed; every non-PK attribute is fully functionally dependent on the PK **3NF**: Any transitive dependencies have been removed **BCNF (Boyce-Codd NF)**: Any remaining anomalies that result from functional dependencies have been removed; every determinant is a CK (candidate key)

```
-- Create the tables
---------------------------
create table Company (
     CName char(20) PRIMARY KEY,
     StockPrice int,
     Country char(20) );

create table Product (
     PName char(20),
     Price decimal(9, 2),
     Category char(20),
     Manufacturer char(20),
 PRIMARY KEY (PName),
 FOREIGN KEY (Manufacturer)
  REFERENCES Company(CName) );
```

Menu served at event, event has work sched, staff in sched, staff supervises staff, dishes in menu

Class participation



PARTICIPATION

I intend to keep our class interactive and ask questions. If I don't have volunteers, I may ask for volunteers from the last rows. If you do not like to participate and be called upon, make sure not sit in the last rows but instead in the front or middle rows.

A rough guide for assigning participation points:

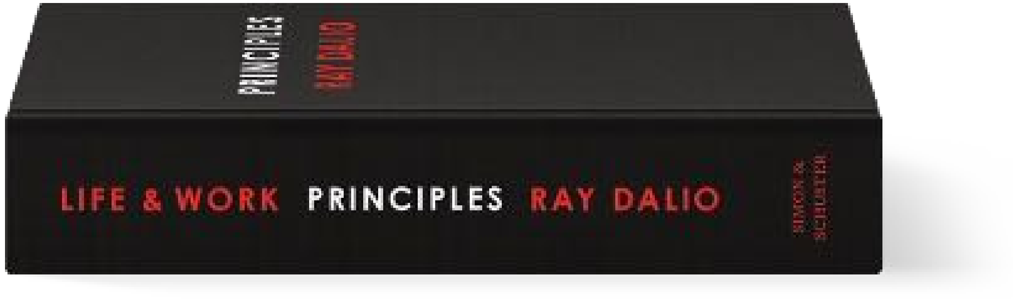
- never asking nor answering questions during class, never answering questions on Piazza = 0 points
- answering questions during class regularly (does not matter whether right or wrong) = 5 points
- answering questions carefully and regularly on Piazza before I can a few times = 5 points
- asking a few very insightful questions during class = 5 points

Rest: somewhere in between

Please add photos to your profiles in Canvas and Piazza so I can more easily remember your class contributions.

A suggestion on how to best use class time!

- It is ok to make mistakes in class. Making mistakes in class is actually the best thing that can happen to you. You learn and will never make it again ☺
- From Ray Dalio's Principles (2017):
  - "Create a Culture in Which It Is Okay to Make Mistakes and Unacceptable Not to Learn from Them"
  - "Recognize that mistakes are a natural part of the evolutionary process."
  - "Don’t feel bad about your mistakes or those of others. Love them!"

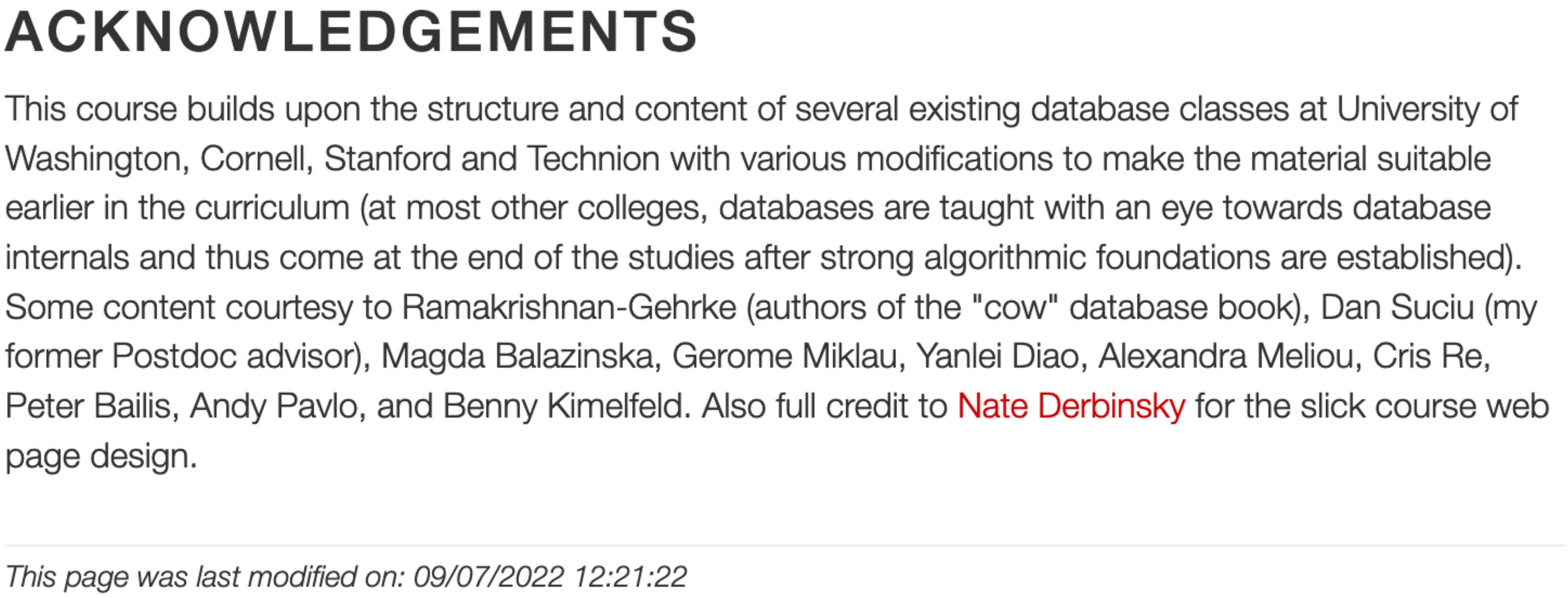


Source: Ray Dalio. https://en.wikipedia.org/wiki/Principles_(book)

Alternative class participation (optional)

- Suggested by a student from an earlier semester who felt he could contribute, but felt too uneasy in the class environment
- Completely optional, at the end of the class, and if worried about participation grade
- You create a few PPTX slides that illustrate any important concept from class in a way that you think explains them *better* than how it was taught in class.
  - Say, you really found outer joins confusing, or translating ERDs into relational schemas, or conflict graphs, or all the logarithm bases in the cost models. Then after you looked deeper, talked to your study mates, all of a sudden it made sense, and you think: "I could have explained that so much better than he did."
- If a student contributes *high-quality* content (i.e. content that I deem so great that I intend to incorporate it to illustrate any concepts next time), then this may slightly increase the class participation.
  - Content that is just a re-hashing of my own material gets no extra points
- Attribute your material

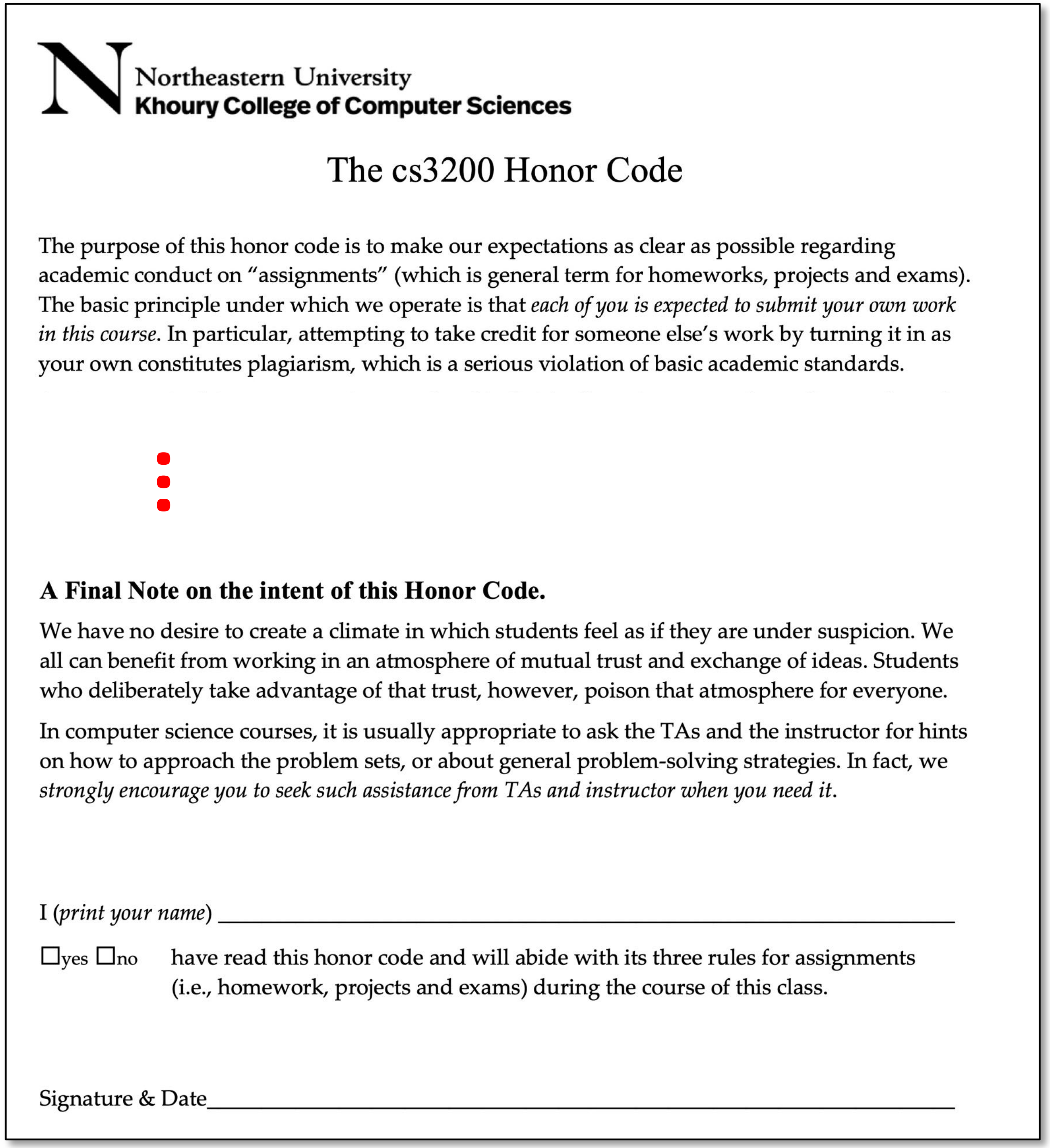
My acknowledgements



ACKNOWLEDGEMENTS

This course builds upon the structure and content of several existing database classes at University of Washington, Cornell, Stanford and Technion with various modifications to make the material suitable earlier in the curriculum (at most other colleges, databases are taught with an eye towards database internals and thus come at the end of the studies after strong algorithmic foundations are established). Some content courtesy to Ramakrishnan-Gehrke (authors of the "cow" database book), Dan Suciu (my former Postdoc advisor), Magda Balazinska, Gerome Miklau, Yanlei Diao, Alexandra Meliou, Cris Re, Peter Bailis, Andy Pavlo, and Benny Kimelfeld. Also full credit to Nate Derbinsky for the slick course web page design.

This page was last modified on: 09/07/2022 12:21:22



Northeastern University
Khoury College of Computer Sciences

The cs3200 Honor Code

The purpose of this honor code is to make our expectations as clear as possible regarding academic conduct on “assignments” (which is general term for homeworks, projects and exams). The basic principle under which we operate is that *each of you is expected to submit your own work in this course*. In particular, attempting to take credit for someone else’s work by turning it in as your own constitutes plagiarism, which is a serious violation of basic academic standards.

⋮

A Final Note on the intent of this Honor Code.

We have no desire to create a climate in which students feel as if they are under suspicion. We all can benefit from working in an atmosphere of mutual trust and exchange of ideas. Students who deliberately take advantage of that trust, however, poison that atmosphere for everyone.

In computer science courses, it is usually appropriate to ask the TAs and the instructor for hints on how to approach the problem sets, or about general problem-solving strategies. In fact, we *strongly encourage you to seek such assistance from TAs and instructor when you need it*.

I (*print your name*) ____

□yes □no have read this honor code and will abide with its three rules for assignments (i.e., homework, projects and exams) during the course of this class.

Signature & Date____

Please don't yet sign. I will make another change before next week

Study groups are great for learning material!

- "... The groups of students who were doing best spontaneously formed study groups...
- Students who were not doing as well tended to do as the instructor suggested-study two hours out of class for every hour in class-but did it by themselves with little social support...
- ... even well-prepared students (high math SATs) are often disadvantaged by high school experiences that lead them to work alone."

Source: Craig Nelson, Dysfunctional Illusions of Rigor: Part 1 - Basic Illusions, http://cgi.stanford.edu/~dept-ctl/tomprof/posting.php?ID=1058

A quizz

Which of the following lowers your measured IQ the most:

A. Smoking marijuana before taking test.

B. Responding to email/texting while taking test.

C. Losing a nights sleep before taking test.

A quizz

Which of the following lowers your measured IQ the most:

A. Smoking marijuana before taking test.

B. Responding to email/texting while taking test.

C. Losing a nights sleep before taking test.

Answer: B

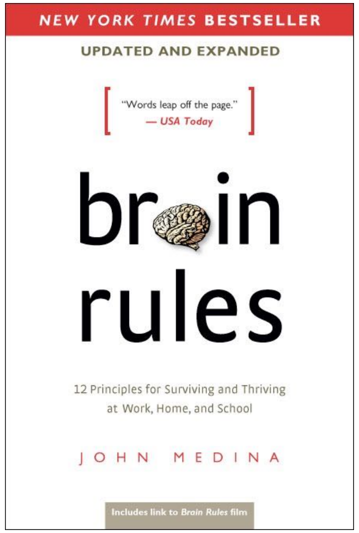
- You suck at multitasking!
- Everyone sucks at multitasking

Source: Courtesy of Michael D Smith (https://mds.heinz.cmu.edu/), http://news.bbc.co.uk/2/hi/uk_news/4471607.stm
(It is a bit of an over-simplification. Clarifications by the original author are here: http://www.drglennwilson.com/Infomania_experiment_for_HP.doc)

Multitasking

“Myth #3: Multitasking when it comes to paying attention, is a myth… studies show that a person who is interrupted takes 50% longer to accomplish a task. Not only that, he or she makes up to 50% more errors” -- John Medina (Brain rules)

“…multitasking is a lie. You’re asking me to switch attention, and that makes me less productive.” -- Dave Crenshaw (The myth of multitasking)



NEW YORK TIMES BESTSELLER

UPDATED AND EXPANDED

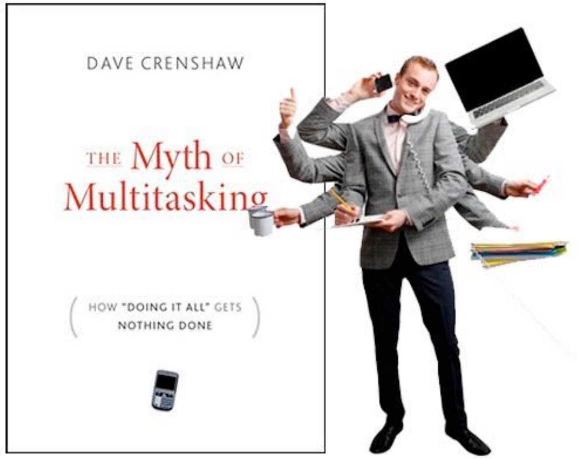
[“Words leap off the page.” — USA Today]

brain rules

12 Principles for Surviving and Thriving at Work, Home, and School

JOHN MEDINA

Includes link to Brain Rules film



DAVE CRENSHAW

THE Myth OF Multitasking

(HOW “DOING IT ALL” GETS NOTHING DONE)

“multitasking adversely affects how you learn. Even if you learn while multitasking, that learning is less flexible and more specialized, so you cannot retrieve the information as easily.” --Russell Poldrack, UCLA Psychology Professor

“Our research offers neurological evidence that the brain cannot effectively do two things at once.” -- Rene Marois, Dept. of Psychology, Vanderbilt

“The brain is a lot like a computer. You may have several screens open on your desktop, but you’re able to think about only one at a time.” -- William Stixrud, Neuropsychologist

Source: Courtesy of Michael D Smith (https://mds.heinz.cmu.edu/)

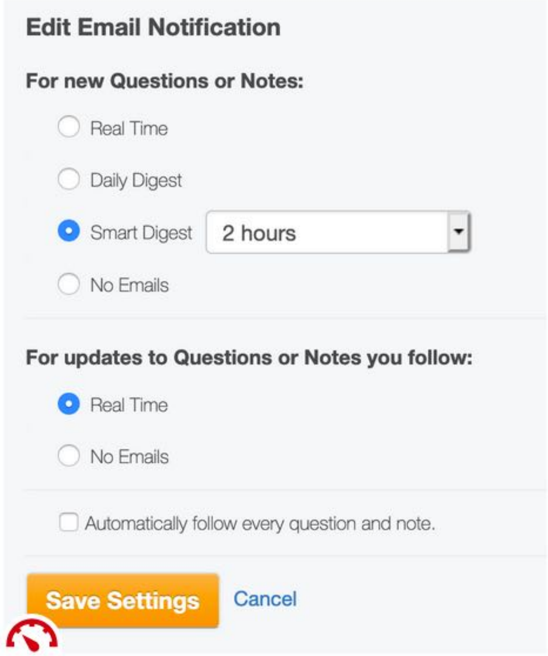
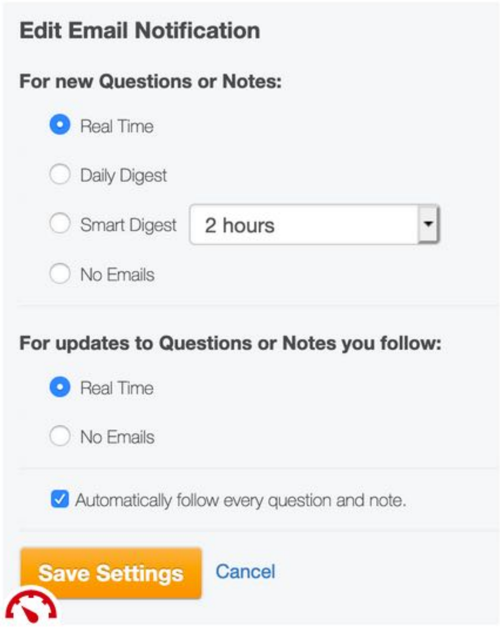
Course evaluation: Thanks for leaving *detailed* Feedback ☺

1. Topics most interesting to you (and why): 1 SQL, 2 DB design 3 Transactions, 4 Internals, 5 NoSQL: more or less material? / what part most difficult / slower or faster?
2. Class organization / Website / : did you find what you were looking for? / what was difficult to find or follow? What would have helped? Suggestions for Canvas or website or Piazza or Gradescope?
3. Installing software: what was difficult to install? What would have helped in addition to the provided PDFs (installation videos?)? Should a future version of cs3200 use Python in connection with databases?
4. ~~Jupyter notebooks: what went well or wrong? how to improve?~~
5. What aspects helped you learn and not forget: more cold calling / group exercises / short slide exercises (SQL) / hands-on SQL typing / FMs on homework solutions / Office Hours / TA Office Hours
6. Learning material: SQL on slides vs. SQL typing/ slides / textbooks / other resources
7. ~~Use of computers & social media in class: yes / no~~
8. HW & group projects: working in groups / assigned random groups / detail of feedback in HW solutions / peer evaluation / slacking off vs. learning
9. Assessment & cheating: more British style: homeworks = practice / final = test
10. Best practice from other classes / what to copy *to* other classes. Other ways I can help: office hours / anonymous feedback form / 5min breaks / 5min "social breaks" where I assign you to talk to somebody
11. How to make you engage more actively?
12. …

Piazza extends our classroom – please subscribe

We use Piazza as our main online message board. If I have updates to share, I will post them on Piazza. Thus I recommend you to automatically follow every and note.

→ *Click on the arrow on the right upper corner from Piazza → Account/Email settings → Edit Email notifications:*

Before	After
 **Edit Email Notification** **For new Questions or Notes:** ○ Real Time ○ Daily Digest ● Smart Digest 2 hours ○ No Emails **For updates to Questions or Notes you follow:** ● Real Time ○ No Emails □ Automatically follow every question and note. Save Settings Cancel	 **Edit Email Notification** **For new Questions or Notes:** ● Real Time ○ Daily Digest ○ Smart Digest 2 hours ○ No Emails **For updates to Questions or Notes you follow:** ● Real Time ○ No Emails ☑ Automatically follow every question and note. Save Settings Cancel

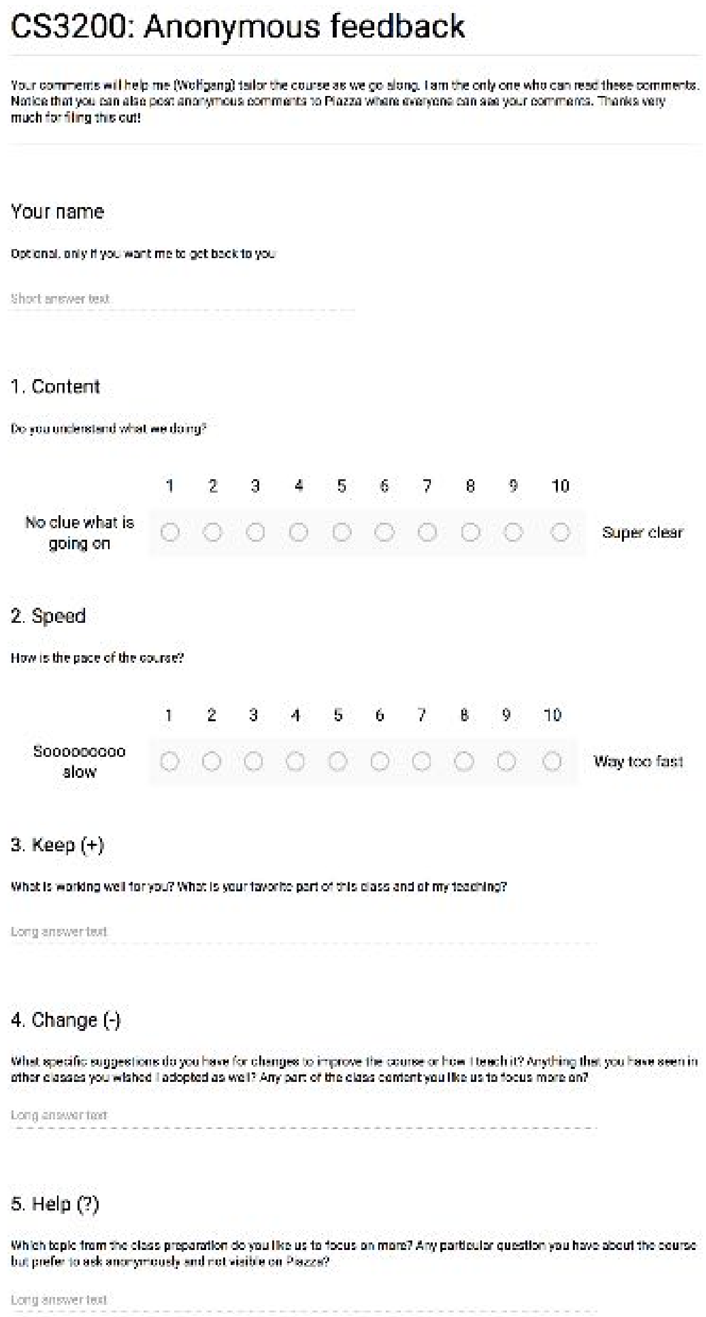
Feedback throughout the semester

Please use this simple way to let me know what works or not!

https://goo.gl/sLJJeH

Even if you find minor annoying issues (spelling mistakes, outdated links, confusing explanations), please spend a minute to let me know. It will improve the rest of your class.

Piazza is visible to everyone in this class. This form only to me



CS3200: Anonymous feedback

Your comments will help me (Wolfgang) tailor the course as we go along. I am the only one who can read these comments. Notice that you can also post anonymous comments to Piazza where everyone can see your comments. Thanks very much for filing this out!

Your name

Optional, only if you want me to get back to you

Short answer text

1. Content

Do you understand what we doing?

	1	2	3	4	5	6	7	8	9	10	
No clue what is going on	○	○	○	○	○	○	○	○	○	○	Super clear

2. Speed

How is the pace of the course?

	1	2	3	4	5	6	7	8	9	10	
Sooooooooo slow	○	○	○	○	○	○	○	○	○	○	Way too fast

3. Keep (+)

What is working well for you? What is your favorite part of this class and of my teaching?

Long answer text

4. Change (-)

What specific suggestions do you have for changes to improve the course or how I teach it? Anything that you have seen in other classes you wished I adopted as well? Any part of the class content you like us to focus more on?

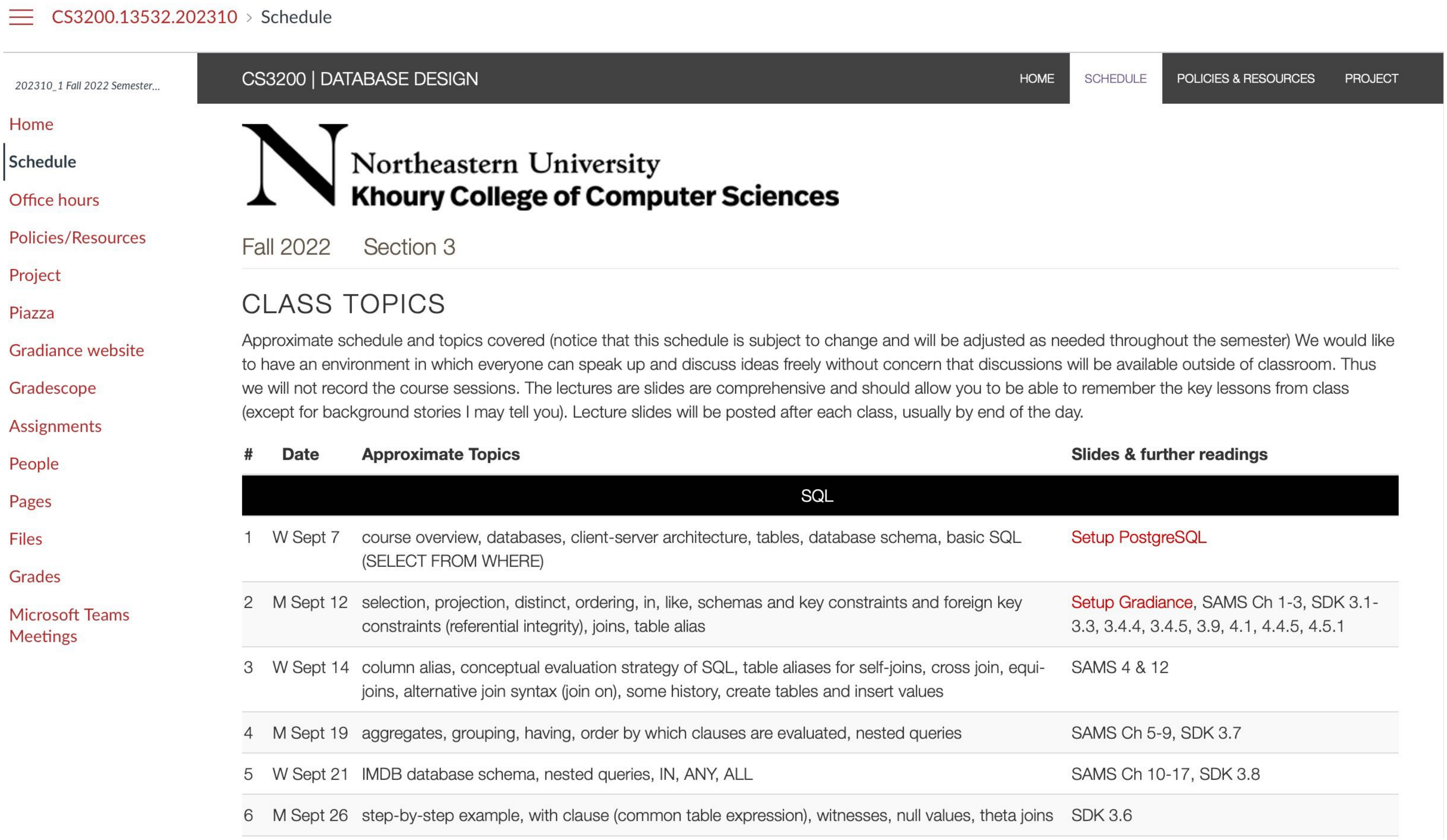
Long answer text

5. Help (?)

Which topic from the class preparation do you like us to focus on more? Any particular question you have about the course but prefer to ask anonymously and not visible on Piazza?

Long answer text

Canvas

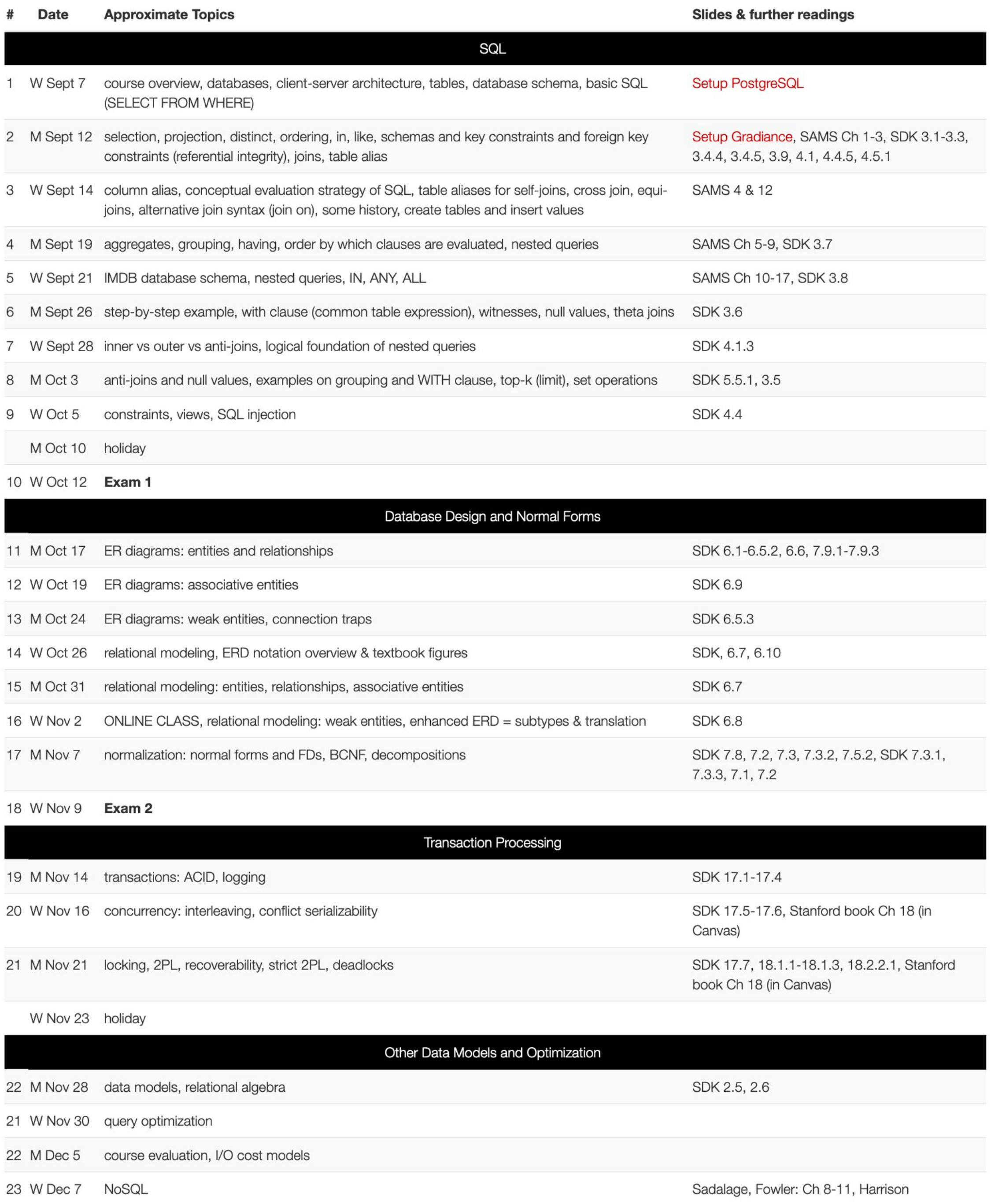


Fall 2022 Section 3

CLASS TOPICS

Approximate schedule and topics covered (notice that this schedule is subject to change and will be adjusted as needed throughout the semester) We would like to have an environment in which everyone can speak up and discuss ideas freely without concern that discussions will be available outside of classroom. Thus we will not record the course sessions. The lectures are slides are comprehensive and should allow you to be able to remember the key lessons from class (except for background stories I may tell you). Lecture slides will be posted after each class, usually by end of the day.

#	Date	Approximate Topics	Slides & further readings
		SQL	
1	W Sept 7	course overview, databases, client-server architecture, tables, database schema, basic SQL (SELECT FROM WHERE)	Setup PostgreSQL
2	M Sept 12	selection, projection, distinct, ordering, in, like, schemas and key constraints and foreign key constraints (referential integrity), joins, table alias	Setup Gradiance, SAMS Ch 1-3, SDK 3.1-3.3, 3.4.4, 3.4.5, 3.9, 4.1, 4.4.5, 4.5.1
3	W Sept 14	column alias, conceptual evaluation strategy of SQL, table aliases for self-joins, cross join, equi-joins, alternative join syntax (join on), some history, create tables and insert values	SAMS 4 & 12
4	M Sept 19	aggregates, grouping, having, order by which clauses are evaluated, nested queries	SAMS Ch 5-9, SDK 3.7
5	W Sept 21	IMDB database schema, nested queries, IN, ANY, ALL	SAMS Ch 10-17, SDK 3.8
6	M Sept 26	step-by-step example, with clause (common table expression), witnesses, null values, theta joins	SDK 3.6



#	Date	Approximate Topics	Slides & further readings
		SQL	
1	W Sept 7	course overview, databases, client-server architecture, tables, database schema, basic SQL (SELECT FROM WHERE)	Setup PostgreSQL
2	M Sept 12	selection, projection, distinct, ordering, in, like, schemas and key constraints and foreign key constraints (referential integrity), joins, table alias	Setup Gradiance, SAMS Ch 1-3, SDK 3.1-3.3, 3.4.4, 3.4.5, 3.9, 4.1, 4.4.5, 4.5.1
3	W Sept 14	column alias, conceptual evaluation strategy of SQL, table aliases for self-joins, cross join, equi-joins, alternative join syntax (join on), some history, create tables and insert values	SAMS 4 & 12
4	M Sept 19	aggregates, grouping, having, order by which clauses are evaluated, nested queries	SAMS Ch 5-9, SDK 3.7
5	W Sept 21	IMDB database schema, nested queries, IN, ANY, ALL	SAMS Ch 10-17, SDK 3.8
6	M Sept 26	step-by-step example, with clause (common table expression), witnesses, null values, theta joins	SDK 3.6
7	W Sept 28	inner vs outer vs anti-joins, logical foundation of nested queries	SDK 4.1.3
8	M Oct 3	anti-joins and null values, examples on grouping and WITH clause, top-k (limit), set operations	SDK 5.5.1, 3.5
9	W Oct 5	constraints, views, SQL injection	SDK 4.4
	M Oct 10	holiday	
10	W Oct 12	**Exam 1**	
		Database Design and Normal Forms	
11	M Oct 17	ER diagrams: entities and relationships	SDK 6.1-6.5.2, 6.6, 7.9.1-7.9.3
12	W Oct 19	ER diagrams: associative entities	SDK 6.9
13	M Oct 24	ER diagrams: weak entities, connection traps	SDK 6.5.3
14	W Oct 26	relational modeling, ERD notation overview & textbook figures	SDK, 6.7, 6.10
15	M Oct 31	relational modeling: entities, relationships, associative entities	SDK 6.7
16	W Nov 2	ONLINE CLASS, relational modeling: weak entities, enhanced ERD = subtypes & translation	SDK 6.8
17	M Nov 7	normalization: normal forms and FDs, BCNF, decompositions	SDK 7.8, 7.2, 7.3, 7.3.2, 7.5.2, SDK 7.3.1, 7.3.3, 7.1, 7.2
18	W Nov 9	**Exam 2**	
		Transaction Processing	
19	M Nov 14	transactions: ACID, logging	SDK 17.1-17.4
20	W Nov 16	concurrency: interleaving, conflict serializability	SDK 17.5-17.6, Stanford book Ch 18 (in Canvas)
21	M Nov 21	locking, 2PL, recoverability, strict 2PL, deadlocks	SDK 17.7, 18.1.1-18.1.3, 18.2.2.1, Stanford book Ch 18 (in Canvas)
	W Nov 23	holiday	
		Other Data Models and Optimization	
22	M Nov 28	data models, relational algebra	SDK 2.5, 2.6
21	W Nov 30	query optimization	
22	M Dec 5	course evaluation, I/O cost models	
23	W Dec 7	NoSQL	Sadalage, Fowler: Ch 8-11, Harrison

- Assignments usually due THU end of day (see Canvas calendar)
- Let me know if the way the calendar in Canvas is currently set up is not working and how to improve

COURSE TEAM & GETTING-IN-TOUCH

Please communicate with us regularly. If you don't understand something, please ask questions! We love questions. One of the benefits of attending a university and interactive classes as opposed to reading a book is that you get to interact with faculty, TAs, and your peers. We are continuously striving to become better and several questions of students who have attended this or similar classes in the past have helped me (Wolfgang) think about some of the illustrative examples you see in class. If at all possible, please reach out to us with your questions via Piazza (linked to from within Canvas), so whoever of us is at hand can answer your question first, plus your peers can see the discussion and join the conversation. You can also leave anonymous feedback via this anonymous Google form, which only I can see. If you want to send me a private message, please use email, not Piazza nor Canvas.

Wolfgang Gatterbauer (Instructor)

E-mail	w.gatterbauer@northeastern.edu
Web	https://gatterbauer.name
Office Hours	directly after class, or Wed 8am-9am @ WVH450 (9/14-12/7, except 11/2 & 11/23), or via Microsoft Teams scheduled by email (in your email to me, please state topic and propose 3 different time slots I can choose from)

Neha Makhija (Head Teaching Assistant)

E-mail	makhija.n@northeastern.edu
Web	https://nehamakhija.github.io/
Office Hours	Tue 3pm-5pm, or Thu 10am-noon

Grishma Rajendra Alshi (Teaching Assistant)

E-mail alshi.g@northeastern.edu

SQL, SQL, SQL…