

L01: Course Overview

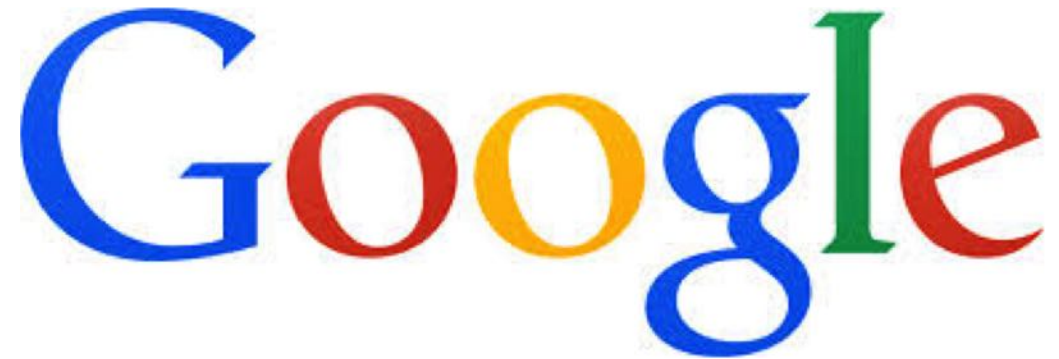
CS3200 Database design (fa18 s2)

<https://northeastern-datalab.github.io/cs3200/>

Version 9/6/2018

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.

Key Questions We Will Answer

- How can we **collect and store** large amounts of data?
 - By building tools and data structures to efficiently index and serve data
- How can we **efficiently query** data?
 - By compiling high-level declarative queries into efficient low-level plans
- How can we **safely update** data?
 - By managing concurrent access to state as it is read and written
- How do different database systems manage **design trade-offs**?
 - e.g., at scale, in a distributed environment?

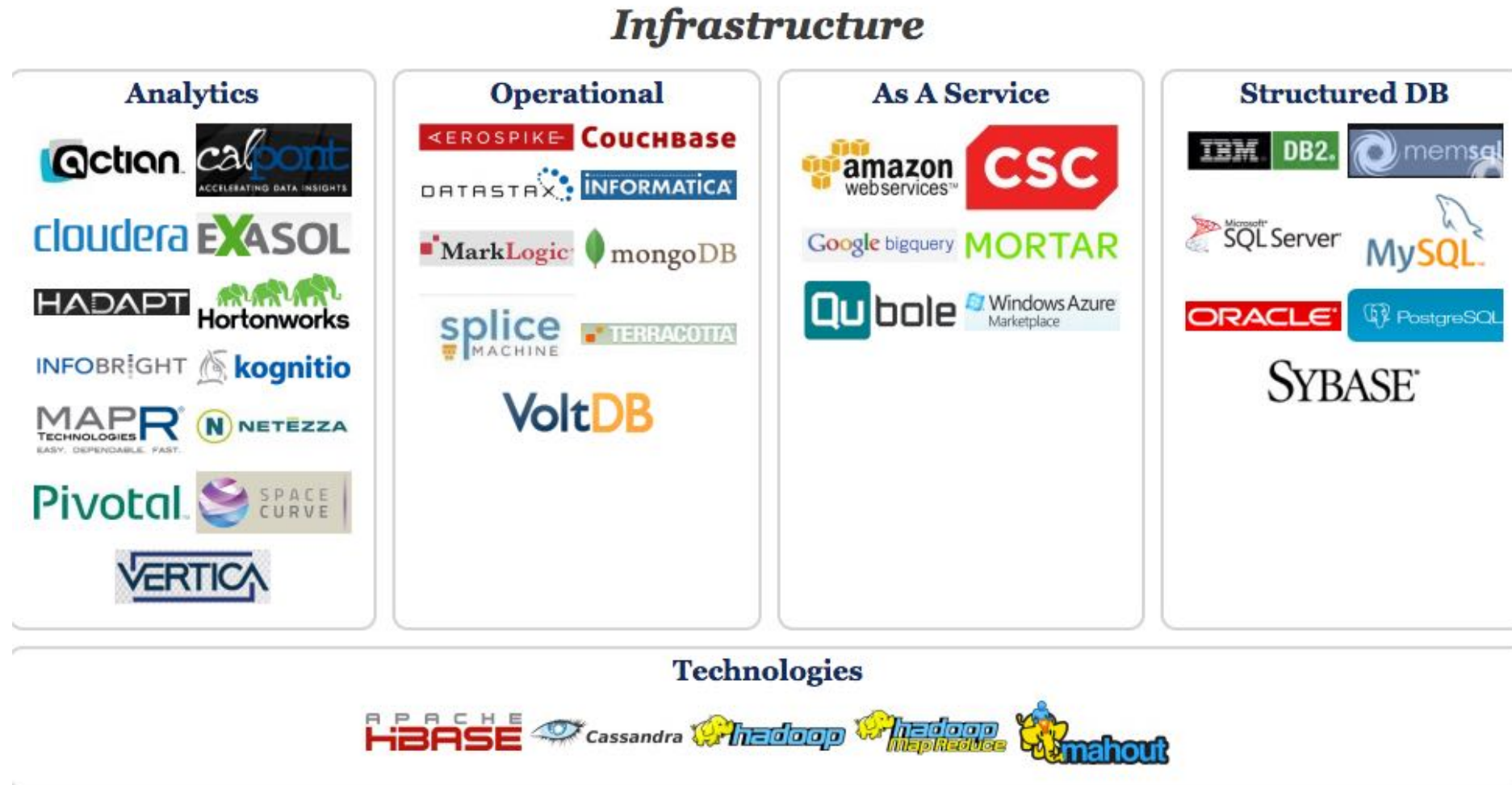
When you'll use this material

- 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

Agenda

1. Introduction, admin & setup, pedagogy
2. Overview of the relational data model
3. SQL, SQL, SQL...

Big Data Landscape... Infrastructure is Changing



New tech. Same Principles.

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

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

Some "birth-years"

- 2004: Facebook
- 1998: Google
- 1995: Java, Ruby
- 1993: World Wide Web
- 1991: Python
- 1985: Windows
- 1974: SQL

Why should you study databases?

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

- Intellectual:

Microsoft

ORACLE



Google

- 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....

N U O R A R O U
D A S U N
W E S D
H O S E M O C
A T G

Many great computer systems ideas started in DB.

What this course is about (and what not)

- Discuss **fundamentals of data management**
 - How to design databases, query databases, build applications with them.
 - How to debug them when they go wrong!
 - Not how to be a DBA or how to tune Oracle 12g or to build a website with MySQL
- We'll cover **how database management systems work**
- And some (but not all of) **the principles of how to build** them

Who we are...

- Instructor (me) Wolfgang Gatterbauer
 - One of three tenure-track faculty in the DATA lab (<https://db.ccis.northeastern.edu/>)
 - Taught before at University of Washington and CMU's business school
 - Research: theoretic foundations for scalable data management
 - Office hours: M 2:00-3:00, WVH 450 (or via email: please suggest 3 time slots)

Teaching Assistants

Niklas Smedemark-Margulies (Head Teaching Assistant)



E-mail smedemark-margulie.n@husky.neu.edu

Disha Sule



E-mail sule.d@husky.neu.edu

Harrison Hur



E-mail hur.ha@husky.neu.edu

Communication w/ Course Staff

- Piazza

The goal is to get you to answer each other's questions so you can benefit and learn from each other.

- Office hours. Poll on Piazza

- MON: 6-9
- MON, THU: 6-8
- MON: 6-8, THU: 6-7, FRI: 5-6

TAs OHs to be listed on the course website!

- By appointment!

Meeting location: 4th floor WVH

Northeastern University

College of Computer and Information Science

Fall 2018

Section 2

CLASS

Time Mondays, Thursdays 11:45am - 1:25pm

Location RH 236 (Richards Hall)

CRN 14677

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 filling 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	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Super clear

2. Speed

How is the pace of the course?

	1	2	3	4	5	6	7	8	9	10	
Soooooo slow	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	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

Important!

- Students with documented disabilities should send in their accommodation letter from the Disability Resource Center at 20 Dodge Hall by the **end of this week** to me.

Lectures

- Lecture slides cover all **essential material**
 - This is your best reference.
 - We are trying to get away from books, but do have pointers and chapter excerpts that will be posted on BB and on the website
- Try to cover same thing in many ways: Lectures, In-class exercises, homework, exams (no shock)
 - Attendance makes your life easier...
- Provide additional textbook chapter excerpts
 - may explain the same concepts in slightly *different* ways

Attendance

- I dislike mandatory attendance... but in the past we noticed...
 - People who did not attend did worse 😞
 - People who did not attend used more course resources 😞
 - People who did not attend were less happy with the course 😞
- My policy: voluntary (to start!) -- reserve right to change
- However, you are graded by "class participation" (more on that later)

Graded Elements

- Gradiance quizzes (10%)
- Class participation (10%)
- Homeworks (30%)
- Three exams (50% = 15% + 15% + 20%)

Homeworks are typically due Mondays end of day, and are posted 1 week before due date (by Mondays after class)

Un-Graded Elements

- Readings provided to just help you!
 - Only items in lecture, homeworks, gradiance quizzes, or in-class activities are fair game.
- In-class activities are mainly to help / be fun!
 - Will occur during class- not graded, but count as part of lecture material (fair game as well)

What is expected from you

- Attend lectures
 - If you don't, it's at your own peril
- Be active and think critically
 - Ask questions, post comments on forums, correct me in class
- Do programming and homework projects
 - Start early and be honest, work with your randomly assigned teams
- Study for exams

Interested in research & graduate school?

- **CS 3950: Introduction to Computer Science Research**
 - new undergraduate class in **Spring'19** by Prof. Dave Choffnes
- Goal: cover the "Science" in "Computer Science"
- Approximate content
 - basic scientific principles and methods
 - selected seminal research in CS
 - key current CS research topics across all major areas of CS
- By the end of the class, students are able to:
 - confidently read and discuss selected research papers
 - have an overview of current hot research topics in CS



Interested in research? Visit our "activities" page

Paper at SIGMOD 2018

- R. Li, M. Riedewald, **Xinyan Deng**
Submodularity of Distributed Join Computation



Poster presentation at Northeast Database day 2018

- R. Li, **Aditya Ghosh**, M. Riedewald, W. Gatterbauer
Optimizing Data Partitioning for Distributed Band Joins
- P. Ojha, **Paul Langton**, W. Gatterbauer
Scalable Compatibility Estimation in Large Network Data



<https://db.ccis.northeastern.edu/activities/>

Or your favorite search Engine: "data lab northeastern"

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

1. **SQL:** Relational data models & Queries

- ~ 5-6 lectures
- How to manipulate data with SQL, a declarative language
 - reduced expressive power but the system can do more for you

2. **Database Design:** Design theory and constraints

- ~ 5-6 lectures
- Designing relational schema to keep your data from getting corrupted

3. **Transactions:** Syntax & supporting systems

- ~ 3-4 lectures
- A programmer's abstraction for data consistency






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

4. **NoSQL**

- ~2-3 lectures
- Key-Value Stores, Column Stores, Document stores, Graph DBs
- (More in CS6240: Large-Scale Parallel Data Processing)

5. **Database internals: Query Processing**

- ~ 3-4 lectures
- Indexing
- External Memory Algorithms (IO model) for sorting and joins
- Basics of query optimization (Cost Estimates)
- Relational algebra

Introduction and SQL				
1	R Sept 6	Course Overview SQL: Introduction		
2	M Sept 10	SQL: Introduction	 Setup SQLite, Setup SQLite (Chrome optional), Setup Gradiance,	Q1
3	R Sept 13	SQL: Intermediate	 SAMS Ch 1-4, 12 Setup PostgreSQL	Q2
4	M Sept 17	SQL: Intermediate	 SAMS Ch 5-9	HW1
5	R Sept 20	SQL: Advanced	 SAMS Ch 10-17 GUW Ch 6	Q3
6	M Sept 24	SQL: Advanced		HW2
Database Design and Normal Forms				
7	R Sept 27	Database Design: ER Diagrams		Q4
8	M Oct 1	Database Design: ER Diagrams		HW3
9	R Oct 4	Exam 1 Database Design: Relations		Q5
	M Oct 8	No class: Columbus Day		
10	R Oct 11	Database Design: Relations		Q6
11	M Oct 15	Database Design: Normalization		HW4
12	R Oct 18	Database Design: Normalization and Decompositions		Q7
Transaction Processing				
13	M Oct 22	Transactions		HW5
14	R Oct 25	Concurrency		Q8
15	M Oct 29	Recovery		HW6
16	R Nov 1	Recovery		Q9

NoSQL				
17	M Nov 5	Exam 2 NoSQL		
18	R Nov 8	NoSQL		Q10
	M Nov 12	No class: Veteran's Day		HW7 (due 11/13)
19	R Nov 15	NoSQL		Q11
Query Processing and Database Internals				
20	M Nov 19	I/O Cost models & Merge Sort		HW8
	R Nov 22	No class: Thanksgiving		
21	M Nov 26	Indexing and B+ trees		HW9
22	R Nov 29	Relational Algebra & Query Optimization		Q12
23	M Dec 3	Course Evaluation, Class Review		HW10, Optional PPTX
	R Dec 6	No class: Reading day		
	Dec 7-14	Exam 3 (time TBD, location: TBD)		

SQL book (other book chapters become available over time)

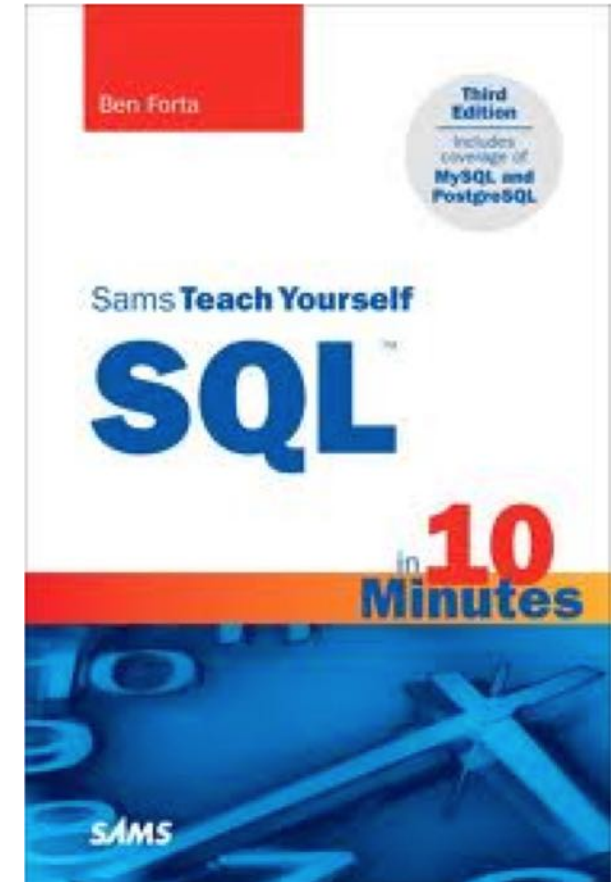
"Sams Teach yourself SQL in 10 minutes" by Ben Forta, 2012 (~\$20)

- Very concise introduction to SQL
- I suggest you read at your own speed
- Online version available from our website

Or:

<http://www.amazon.com/Sams-Teach-Yourself-Minutes-Edition/dp/0672336073>

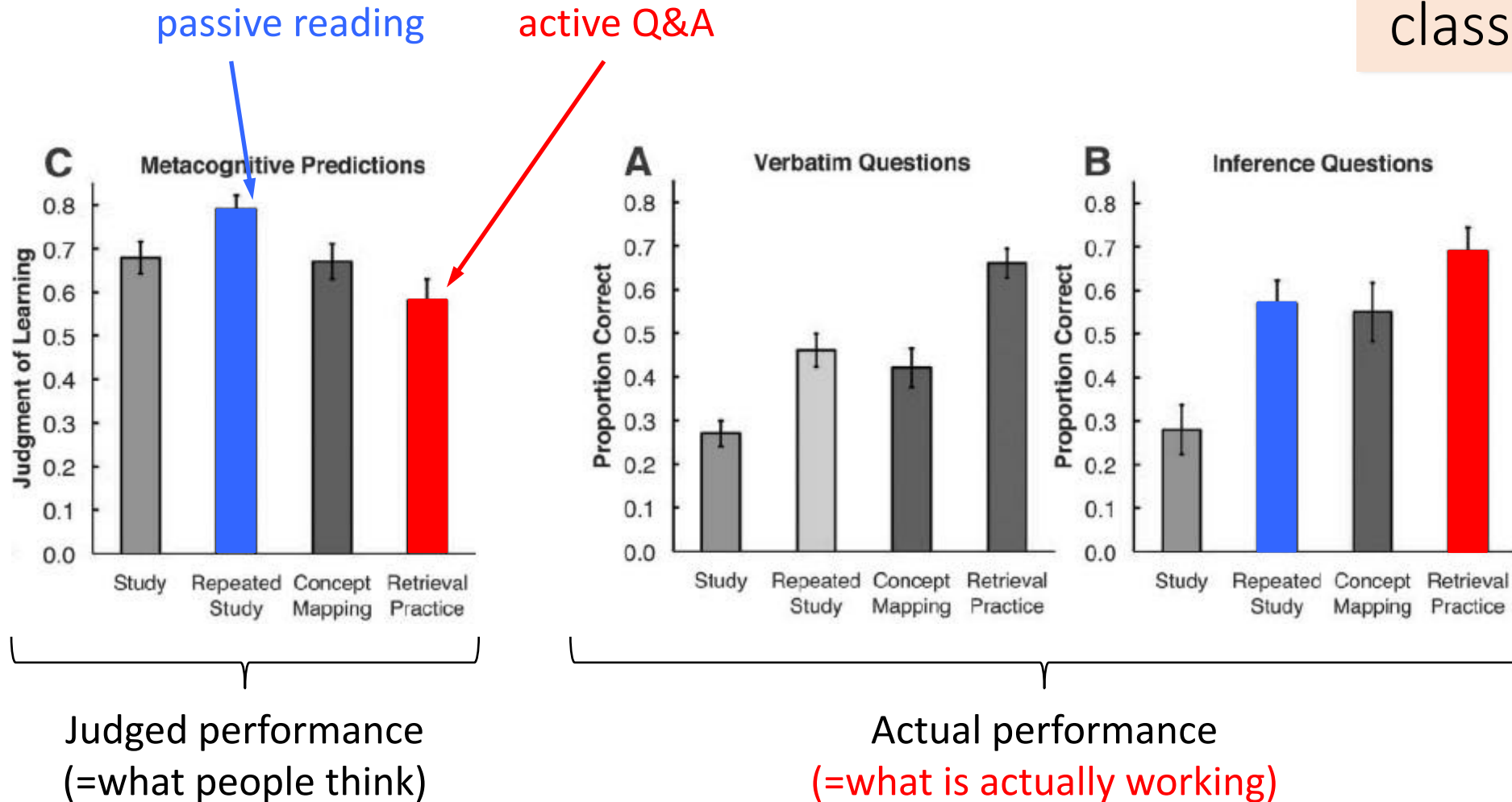
<http://ptgmedia.pearsoncmg.com/images/9780672336072/samplepages/0672336073.pdf>



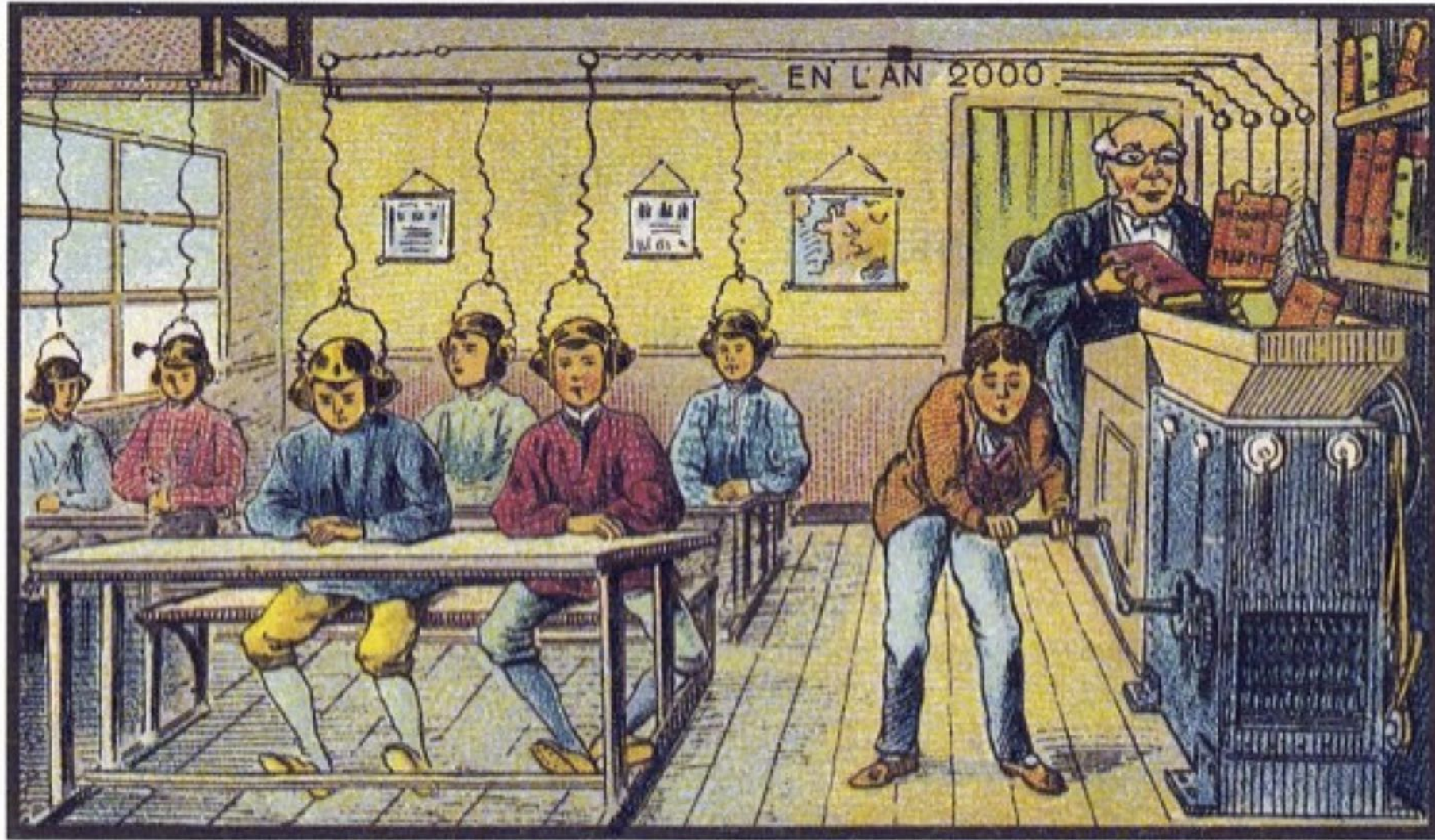
Course Pedagogy

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

⇒ name plates,
class participation

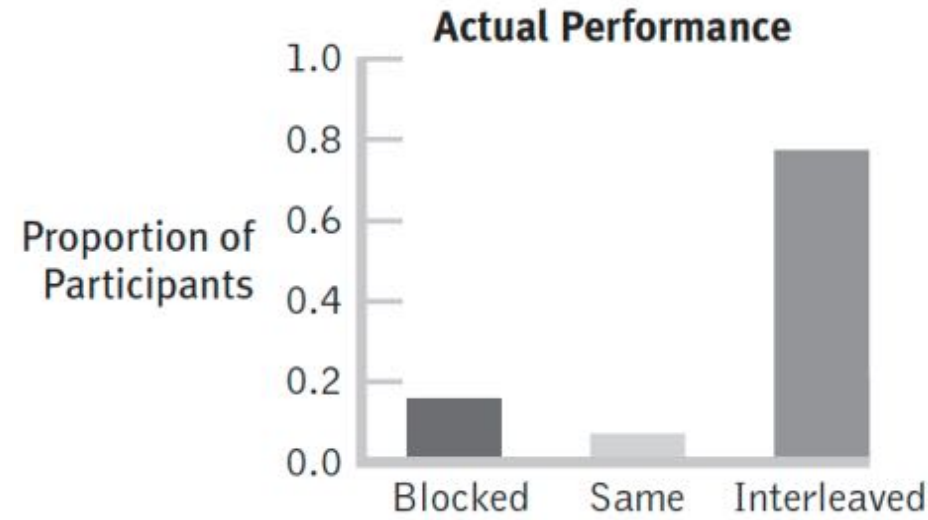
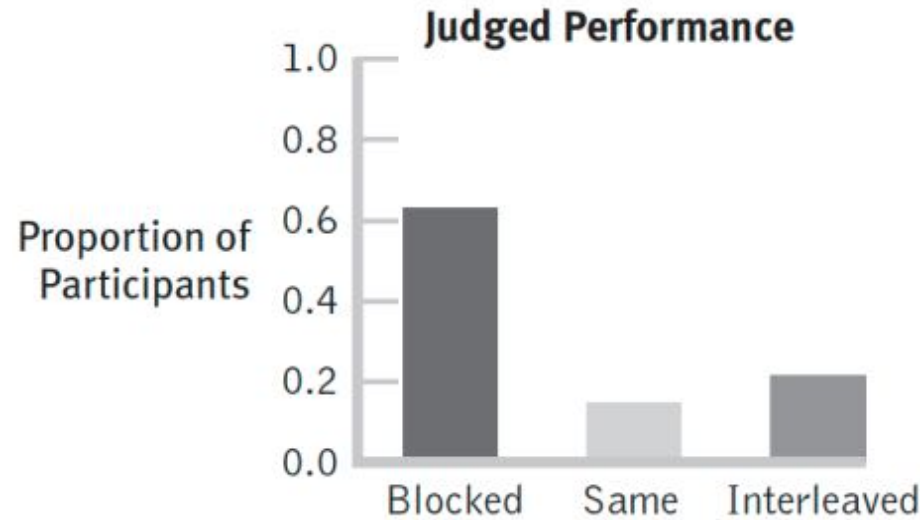


The year 2000 imagined in 1900



At School

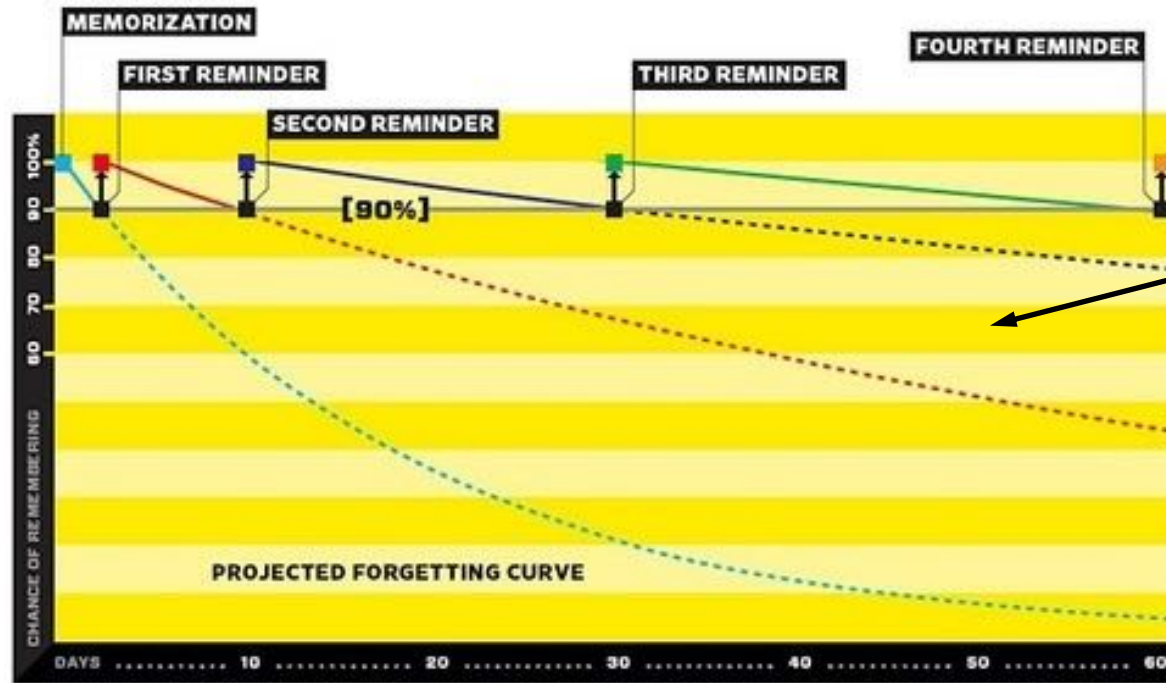
Sequencing Material: "Under which teaching condition do you think you learn better?"



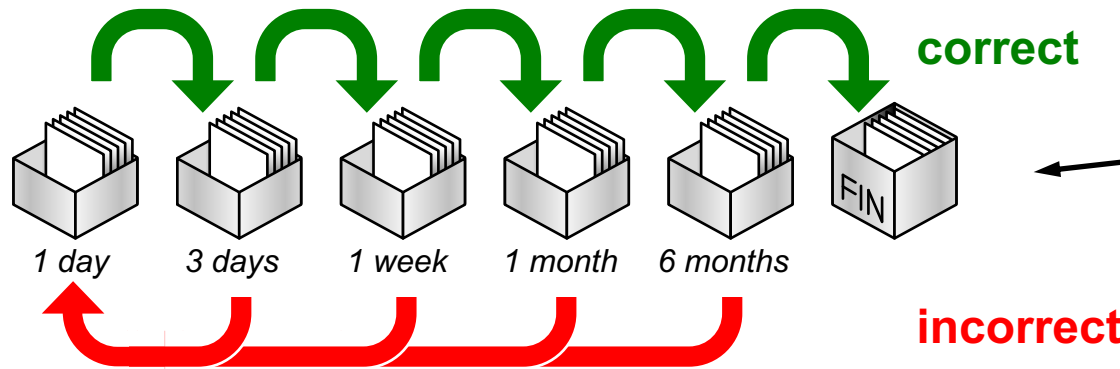
⇒ exams are comprehensive

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.

Spaced Repetition



Ebbinghaus Forgetting Curve



Leitner System
(Pimsleur's graduated interval recall)

The "Surfer Analogy" for time management



Why I don't post slides *before* lecture

From the Preamble of one of the best physics books: „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.

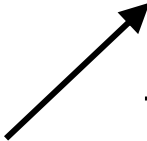
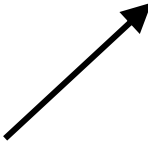
...

...

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

My pedagogic goals for classroom effectiveness

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

Final exam

- Closed book, individual, hard, comprehensive, time-constrained
- Therefore: fair!
- See BB for last exam3
- But don't mistake the dominant group-work throughout the term with you knowing the material alone
- You can bring your own one page scribble page

70-455 FINAL SCRIBBLE BY

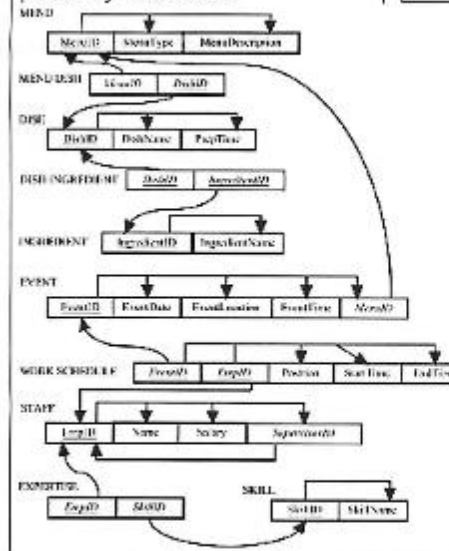
You can write down notes and ideas on this single sheet for your midterm. Fine print: (1) you need to use font size 12, (2) everything needs to fit within the box below, and (3) you need to hand-in this sheet after the midterm, with your andrew ID and name filled in on top.

```
=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 FD have 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) );

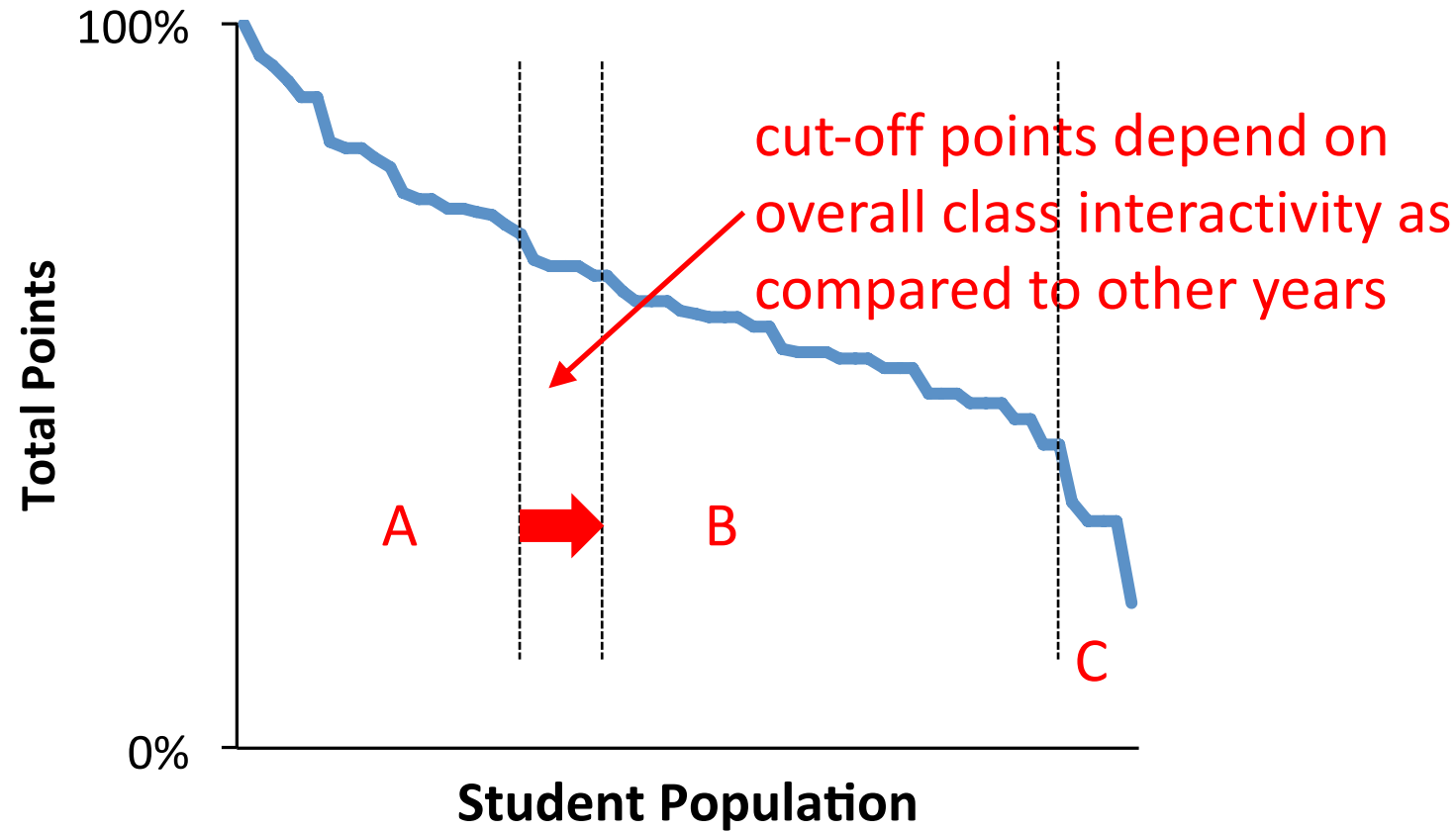
create table Product (
  PName char(20),
  Price decimal(9, 2),
  Category char(20),
  Manufacturer char(20),]
PRIMARY KEY (PName),
FOREIGN KEY (Manufacturer)
REFERENCES Company(CName) );
```

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

Grading Philosophy

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

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



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

Class participation

Participation in class

- Never attends class
- Sometimes attends class
- Attends class regularly
- Attends class regularly & always brings name tag
- Comes to class sometimes, brings name tags, and asks or answers questions regularly
- Comes to class sometimes, brings name tags, and asks questions that make me think and create new slides to answer the question next time

Participation grade

- 0
- 0
- 0
- 0
- 9
- 10

Participating in a large class



Alternative class participation (optional)

- Suggested by a student last 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 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.
 - High-quality: Content that is just a re-hashing of existing material gets no extra points
- Attribute your material

My acknowledgements

- This course builds upon the structure and content of several existing database classes. Some content courtesy to Ramakrishnan-Gehrke, Dan Suciu, Magda Balazinska, Gerome Miklau, Yanlei Diao, Alexandra Meliou, Cris Re, Peter Bailis, Andy Pavlo, and Benny Kimelfeld. Also full credit to Nate Derbinsky and Jan-Willem van de Meent for the slick web page design.

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."

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 Mike Smith, 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)

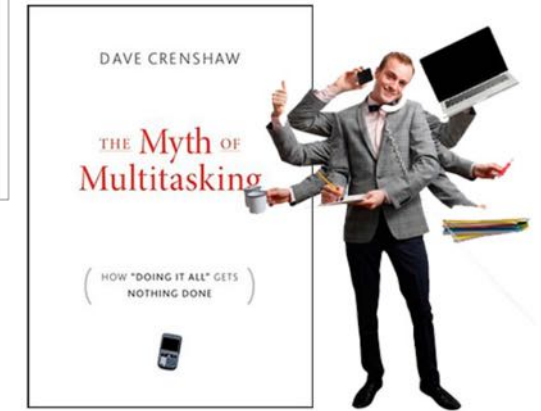
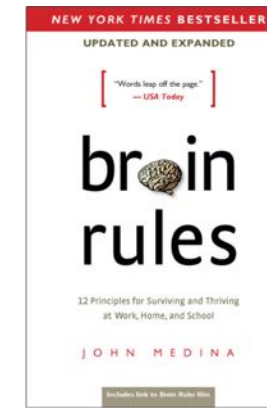
“...multitasking is a lie. You’re asking me to switch attention, and that makes me less productive.” -- Dave Crenshaw (The myth of multitasking)

“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


*If you do something else in class → I will pick on you:
You need to prove to me that you can multitask.*



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 Blackboard 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?) Replace Postgresql with MySQL (Postgresql is better for optimizing, but we won't cover this in detail next time)? Virtual machines?
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? SQL worked really well. More random calling from class list?
- 12....

Ideas for from last semester for this semester

- No project, but more longer homeworks in larger groups
 - keep soft graded HWs, hard exams
- Gradescope, some autograded Jupyter notebook checks for code
- Topics:
 - 1 SQL: no change
 - 2 Database design: shorten and completely replace the Stanford arrow notation with crow foot / UML; personalize Gradiance
 - 3 Transactions: extend and include hands-on exercises
 -  – 4 Database internals: shorten; remove advanced joins, but keep indices and RA
 - 5 NoSQL: extend with hands-on with all 4 types of NoSQL databases;
 - To be decided: all in Jupyter, or actual installations, or preinstalled on VMs

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 filling 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	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Super clear

2. Speed

How is the pace of the course?

	1	2	3	4	5	6	7	8	9	10	
Soooooo slow	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	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

Agenda

1. Introduction, admin & setup, pedagogy

2. Overview of the relational data model

3. SQL, SQL, SQL...

What we will learn about next

1. Definition of DBMS
2. Data models & the relational data model
3. Schemas & data independence

What is a DBMS?

A Database Management System (DBMS) is a piece of software designed to store and manage databases

- A large, integrated collection of data
- Models a real-world enterprise
 - Entities (e.g., Students, Courses)
 - Relationships (e.g., Alice is enrolled in 145)

A Motivating, Running Example

- Consider building a course management system (CMS):

- Students
- Courses
- Professors

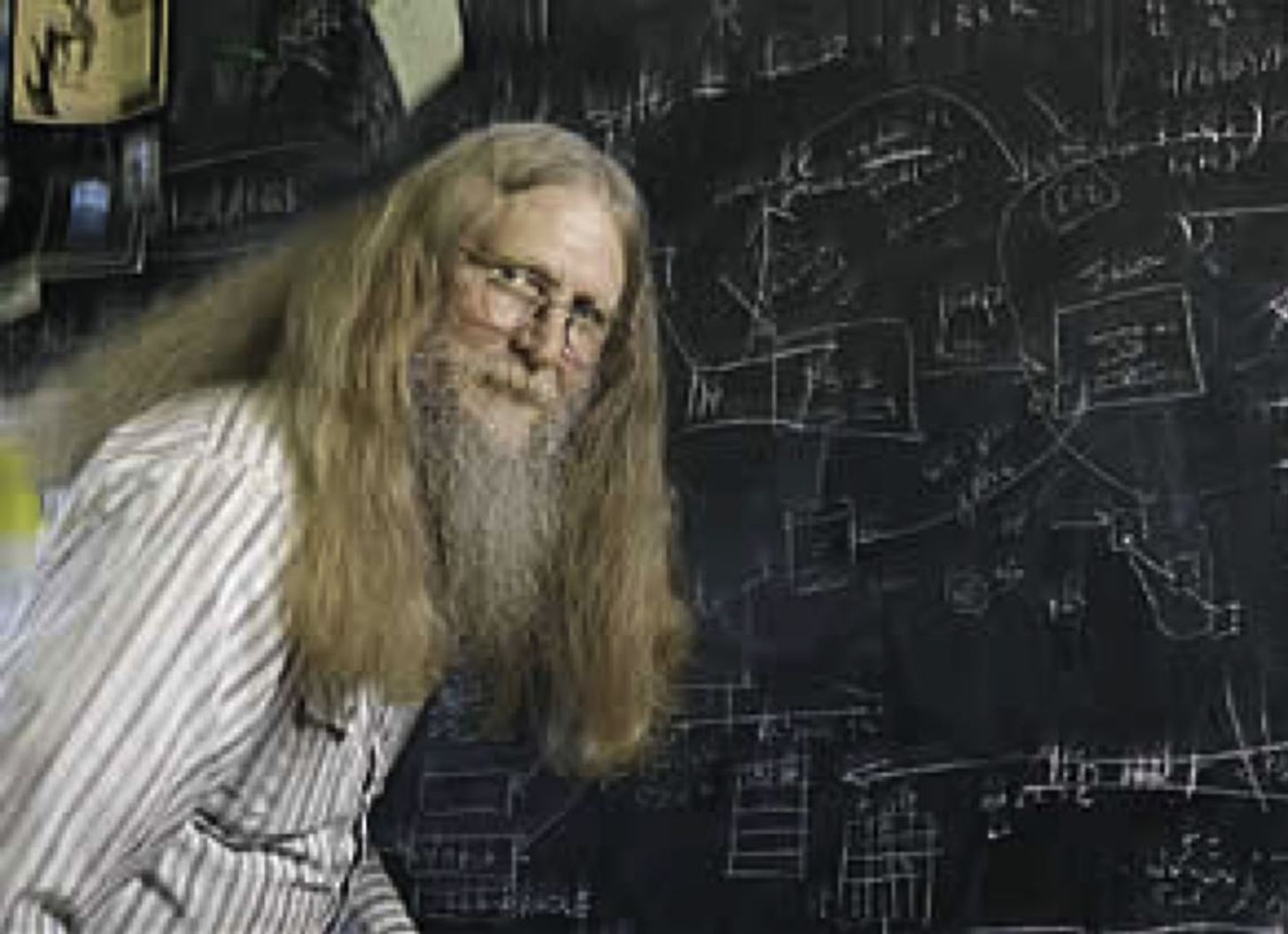
} *Entities*

- Who takes what
- Who teaches what

} *Relationships*

Data models

- A **data model** is a collection of concepts for describing data
 - The relational model of data is the most widely used model today
 - Main Concept: the relation- essentially, a table
- A **schema** is a description of a particular collection of data, **using the given data model**
 - E.g. every relation in a relational data model has a schema describing types, etc.



“Relational
databases are the
foundation of
western
civilization”

Bruce Lindsay, IBM Research

As quoted in: <https://dl.acm.org/citation.cfm?id=1083803>

Modeling the CMS

- Logical Schema
 - Students(sid: string, name: string, gpa: float)
 - Courses(cid: string, cname: string, credits: int)
 - Enrolled(sid: string, cid: string, grade: string)

sid	Name	Gpa
101	Bob	3.2
123	Mary	3.8

Students

Relations

sid	cid	Grade
123	564	A

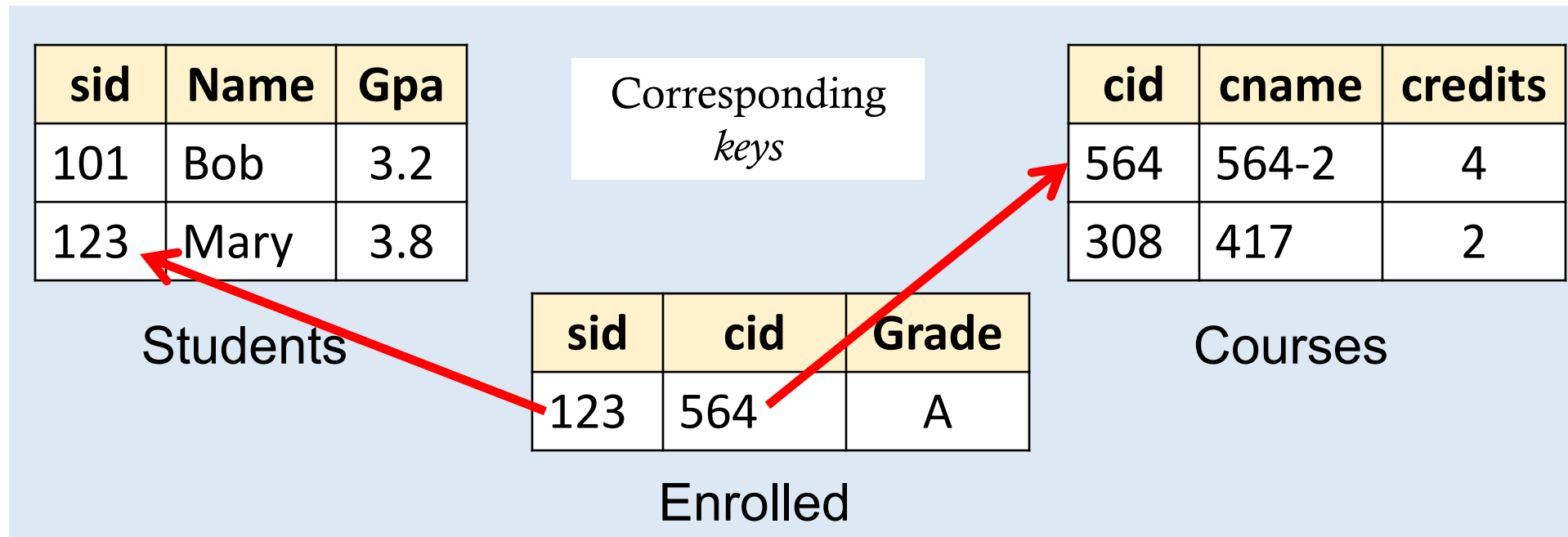
Enrolled

cid	cname	credits
564	564-2	4
308	417	2

Courses

Modeling the CMS

- Logical Schema
 - Students(sid: string, name: string, gpa: float)
 - Courses(cid: string, cname: string, credits: int)
 - Enrolled(sid: string, cid: string, grade: string)



Other Schemata...

- **External Schema:** (Views)
 - Course_info(cid: string, enrollment: integer)
 - Derived from other tables
- **Logical Schema:** Previous slide
- **Physical Schema:** describes data layout
 - Relations as unordered files
 - Some data in sorted order (index)



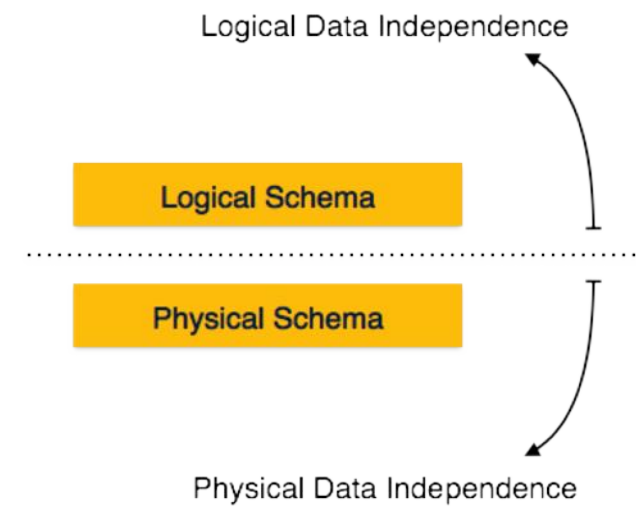
Applications



Administrators

Data independence

- Concept: Applications do not need to worry about how the data is structured and stored



Logical data independence:

protection from changes in the *logical structure of the data*

I.e. should not need to ask: can we add a new entity or attribute without rewriting the application?

Physical data independence:

protection from *physical layout changes*

I.e. should not need to ask: which disks are the data stored on? Is the data indexed?

One of the most important reasons to use a DBMS

Summary of DBMS

- DBMS are used to maintain, query, and manage large datasets.
 - Provide concurrency, recovery from crashes, quick application development, integrity, and security
- Key abstractions give **data independence**
- DBMS R&D is one of the broadest, most exciting fields in CS. Fact!

Agenda

1. Introduction, admin & setup, pedagogy
2. Overview of the relational data model
3. SQL, SQL, SQL...