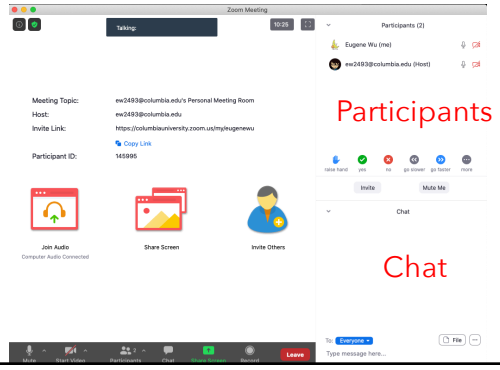
Setup Zoom to Look Like This



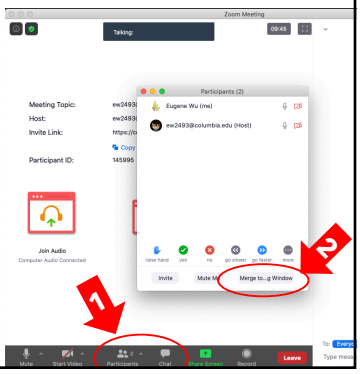
1

Using Zoom

1. Open Chat & Participants
2. Merge Window

Click the green check in the participants list when done

yes



2

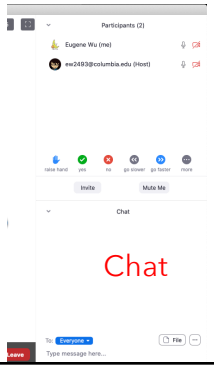
Participation Protocol

Raise Hand to speak
Share thoughts and questions in chat
Thank/applaud colleagues that speak up/present

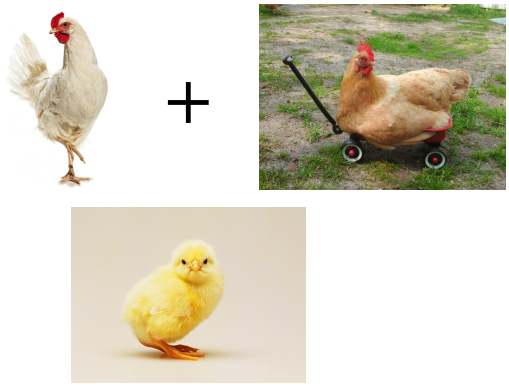
3

Is it possible for
1 + 1 = 3??? If so, when?

Open chat and share what you think



4



5

Participation Protocol

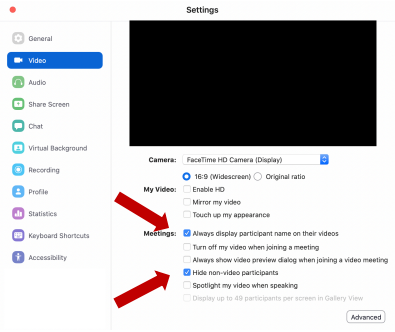
Raise Hand to speak
Share thoughts and questions in chat
Thank/applaud colleagues that speak up/present

I may ask individuals to share thoughts on the readings

6

Using Zoom

Give a green check when done

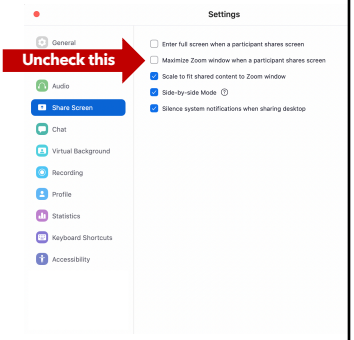


yes

7

Using Zoom

Give a green check when done



yes

8

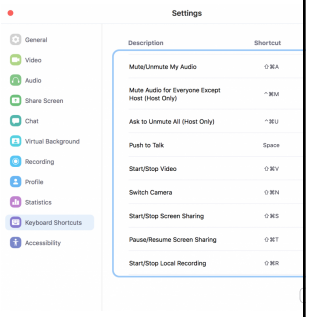
Using Zoom

Raise hand ⌥y

Enable Mic space

Toggle video ⌘⇧V

Toggle chat ⌘⇧H



9

Using Zoom

Raise hand ⌥y

Enable Mic space

Toggle video ⌘⇧V

Toggle chat ⌘⇧H

Turn your video on, enable the Mic, and say

"Hi, my name is _____"

10

COMS W6113

Topics in Database Research

Eugene Wu
http://w6113.github.io

11

Database Research

Declarative API to manage data

Defines and enforces clear semantics
Model the data, query, transactions, hardware, etc

Combines PL, systems, theory, optimization, ML to
Maintain data integrity (under failures)
Run queries correctly and quickly
Support application needs

12

Database Research

Definition of data changing
Records
Graphs
Sensor events
Images & Videos
Documents & PDFs
VR/AR…

Application needs are growing
Hardware, scalability, ML, real-time
Decision making, data exploration, video analytics, etc

13

This Course

Learn about Database Research
Read classic and modern papers
Work on a research project

Learn how to read papers
Problem selection
Insights and technical contributions
Presentation

14

Course Topics

Query Execution
Query Optimization
Columnar Stores
Query Compilation
Large-scale Dataflow
Materialized Views
Datalog and Lineage
Physical database optimization
<Your interest here>

15

Course Topics

We will NOT cover material in 4111, namely

Relational Algebra
SQL
Basics of query execution and optimization
Access methods
Basic design of disk-based database systems

16

What you will do

Project (65%)
Groups of 2-3

Assignments (15%)
See how pieces of a DB engine fit together

Paper Reviews (10%)
Answer questions for each paper & submit review on wiki
Lots of reading!

Class discussion (10%)

17

Project

2 Broad Categories

Reproduce and extend

Research project

18

Reproduce (& Extend)

Reproduce
Read and deeply understand a data management topic
~5 to10+ papers, summarize & compare them
Implement best/combined technique in DataBass/another sys
Benchmark it

& Extend (bonus)
Extend the implementation to make it a bit better
Benchmark to show it is better
Explain when and why the extension is effective

19

Research Project

Investigate new ideas and solutions
Define hypothesis
Conduct research
Evaluate hypothesis
Writeup and research

Emphasis on hypothesis and evaluation
Translate contributions into testable hypotheses
Evaluation should evaluate hypotheses

20

Project Overview

Choose from list of projects, or come up with your own

Pick partners, 1 page proposal
Problem you are solving
Hypothesis
Plan of attack, with milestones
Preliminary related work

Demo/Presentation session

Project report

21

Project Timeline

We are here to help you succeed

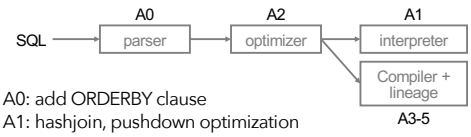
Week
- 02 Release list of projects
- 04 Submit proposal
  - Discuss with staff before this stage
- 07 Submit paper draft
  - Position relative to state of the art
  - Play with related tools
- 11 Check in
- 14 Showcase
- 15 Submit paper 8-10 pgs

22

Programming Assignments

DataBass: Python DB engine ~4K loc 

Hands-on experience with query from parsing to execution



A0: add ORDERBY clause
A1: hashjoin, pushdown optimization
A2: selinger join optimization
A3: hashjoin compilation
A4: benchmarking
A5: lineage

23

Class Format

Submit paper review before class
Quiz for self assessment
Breakout to discuss quiz/reading
Go over paper ideas
Use chat to ask/answer questions

Optional: sign up to lead discussions
Optional: scribe the discussion, add to course wiki

24

Reviews

Add to course wiki
Can skip 4 **paper reviews** (note: 1+ papers per class)
Late submissions not accepted

Questions
What is the problem?
Why does prior work fail to solve it?
Main insight and technical contributions?
Do the evaluations hold up?
What do you wish could be improved?

25

Reviews

Do not plagiarize

Write original reviews
Do ***not*** copy text from papers, online, or other reviews

See http://www.cs.columbia.edu/education/honesty

26

Presentation

15-20 minutes + discussion

Cover **key elements** of paper
- Find & read related work to provide context
- Intuition >> formulae
- Create your own examples. Plagiarism rules apply

Lead discussion with questions

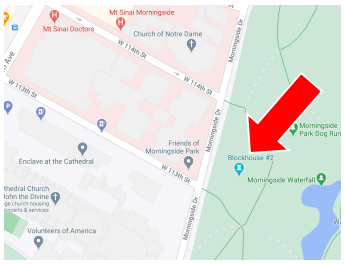
Go over presentation with Eugene ahead of time
Send slides to TA 2 days before class, by midnight.

27

Recitation

Tues 2-3PM EST, an open air space (113th & Morningside)
Attendance is *not* taken

Purpose
Go over examples
Research advice
Life advice
Stay safe from COVID



28

Who am I?

Eugene Wu
421 Mudd (DSI space)
ewu@cs.columbia.edu
OH: Tues 3-4PM EST

PhD MIT, 2015
Started @Columbia Fall 2015
Systems for Human Data Interaction

29

Deka Auliya Akbar

TA for the class
OH: tba

30

Logistics

w6113.github.io

Class	T/Th 11:30 – 1PM, Zoom
Recitation	Tues 2-3PM, open air location TBD
Commun.	Slack
Reviews	Course wiki

Academic Honesty

See http://www.cs.columbia.edu/education/honesty
Ask if unsure, falling behind, etc
Don't plagiarize. Worse than failing.

31

Prereqs

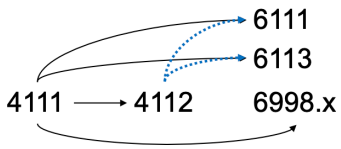
Required: 4111 Intro to DB
Declarative languages and data independence
Relational algebra, SQL, relational model
Query optimization and execution

Useful: 4112

Great to have people with different backgrounds
Talk to me if unsure
If graduate student from different area, talk to me

32

Relationship with other DB classes



33

Enrollment

~25 students
Doesn't matter if enrolled or waitlisted

Admission based on quality of
- First reviews
- Participation in initial classes

34

Grading

Project	65%
Assignments	15%
Reviews	10%
Participation	10%

No curve, everyone can get A
If project is amazing, automatic A

35

Breakout: Getting to know each other

Assign one note taker to summarize the breakout to the class

Introduce yourself

Your name, year, department
What is a database you recently used (maybe indirectly)?
Share where you want to travel when you are able to.

36

5 min Break

37

Short DB History

What Goes Around Comes Around

Michael Stonebraker
Joseph M. Hellerstein

Abstract

This paper provides a summary of 35 years of data model proposals, grouped into 9 different eras. We discuss the proposals of each era, and show that there are only a few basic data modeling ideas, and most have been around a long time. Later proposals inevitably bear a strong resemblance to certain earlier proposals. Hence, it is a worthwhile exercise to study previous proposals.

In addition, we present the lessons learned from the exploration of the proposals in each era. Most current researchers were not around for many of the previous eras, and have limited (if any) understanding of what was previously learned. There is an old adage that he who does not understand history is condemned to repeat it. By presenting "ancient

38

What is a Database?

Data stored in a representation
Records and relationships

Query the data
Write or generate algorithms to traverse storage representation

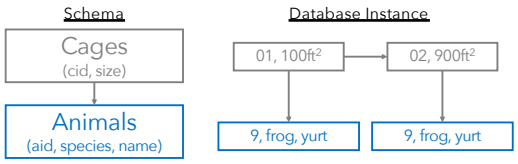
Innovations found in
Data representation
Query interface
Translating query interface → algorithms over data

39

60s Hierarchical Model

IMS: **I**BM **M**anagement **S**ystem
For Apollo space program: Saturn V inventory
Still used by banks today

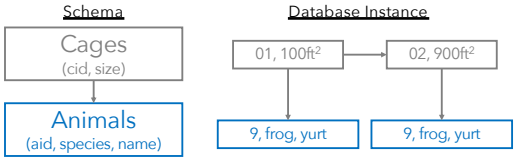
Each level is a different entity type



40

60s Hierarchical Model

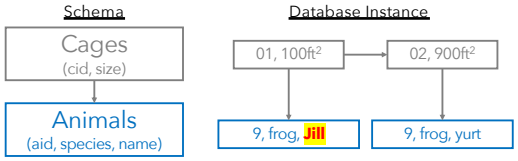
```
# Animals living in cage 2
Find cage where cid = 2
Find 1st child of current record
Loop
     Find next sibling record of same type
     print record
```



41

60s Hierarchical Model

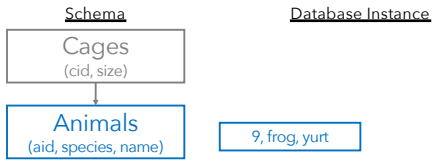
👎 Data redundancy if many to many relationship



42

60s Hierarchical Model

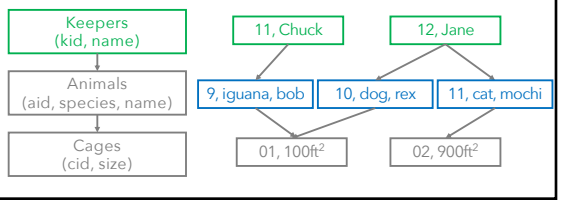
👎 Data redundancy if many to many relationship
👎 Animal can't exist without a cage
👎 Removing cage removes animals



43

60s Network Model (CODASYL)

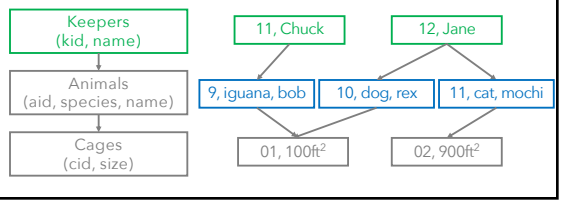
Charles Bachman Turing #8
Record at a time navigation
Impossible to program
Changing data representation breaks programs



44

60s Network Model

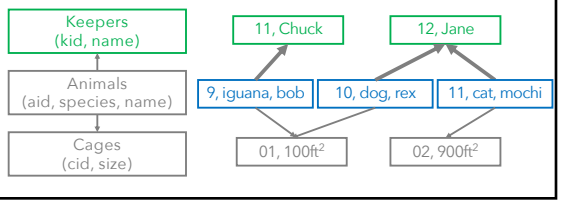
```
Find keeper where name = Jane
Loop
     Find next animal in cares_for
     Find Cage in lives_in
     print record
```



45

60s Network Model

```
Find keeper where name = Jane
Loop
     Find next animal in cares_for
     Find Cage in lives_in
     print record
```

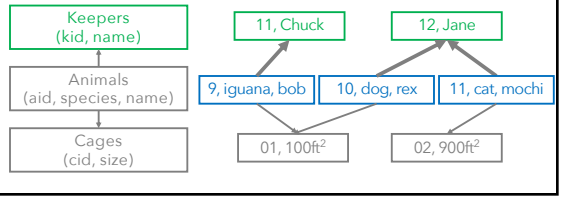


46

60s Network Model

```
Find keeper where name = Jan
Loop
     Find next animal in cares_for
     Find Cage in lives_in
     print record
```

Broken

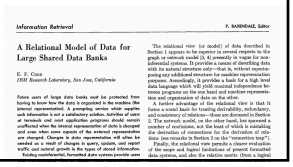


47

70s Relational Model

Edgar Codd Turing #18
Started the relational DB research field
Game changer

Set at a time language
Declarative language
Data independence


Information Retrieval — P. BAXENDALE, Editor

A Relational Model of Data for Large Shared Data Banks

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

48

70s Relational Model

Keepers

kid	name
11	Chuck
12	Jan

Cages

cid	size
1	100ft²
2	900ft²

Cares_for

kid	aid
11	9
12	10
12	11

Lives_in

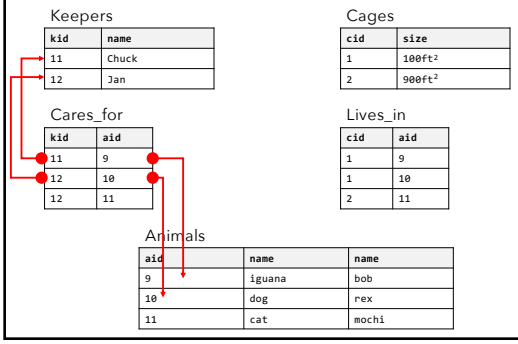
cid	aid
1	9
1	10
2	11

Animals

aid	name	name
9	iguana	bob
10	dog	rex
11	cat	mochi

49

70s Relational Model



50

70s Relational Model

When do you want data independence?

δApp << δenvironment
- Hellerstein

Application logic evolves slowly
Technology evolves rapidly
- Change data layout
- Change schema
- Change/upgrade hardware
- Data system is remote
- Cloud
- …

51

70s Will it run?

Ingres @Berkeley
Stonebraker Turing #32 and Wang
Begat *Ingres, Bitton-lee, Sybase, MS SQLServer, PACE, Tandem*

System R @IBM
Jim Gray Turing #22
Big research team, 15 PhDs
Begat *DB2, Oracle, HP Allbase, Tandem*

Proved that theory is practical
Spawned RDBMS market and modern data-oriented software

52

80s Will it Sell?

Heavyweights
IBM doesn't want to cannibalize IMS
Oracle reads System R papers, and goes to market first
IBM eventually releases DB2
IBM chooses SQL, Oracle follows, becomes the standard

Crowd of players
Jim Gray et al join Tandem
Stonebraker makes Ingres → Informix
Britton & Lee from Ingres create Britton-Lee
Epstein leaves BL to create SyBase (bought by SAP for $5B)
Wang creates PACE
Caltech alums start Teradata w/ networking tech → parallel DB

53

80s Object Oriented DBMS

Impedance mismatch between Objects and relational model

```
Struct person {
  name string;
  hobbies string[];
}
```

person

pid	name
1	Chuck
2	Jane

personhobbies

pid	hobby
1	cheese
1	surfing

Translation cumbersome
Wanted to avoid the join
Prefer DOT notation e.g., person.hobbies

54

80s Object Oriented DBMS

Impedance mismatch between Objects and relational model

```
Struct person {
  name string;
  hobbies string[];
}
```

person

pid	name	hobbies
1	Chuck	[cheese, surfing]
2	Jane	

```
{
  pid: 1
  name: "Chuck"
  hobbies: [
    "cheese",
    "surfing"
  ]
}
```

Nested Relational Model

55

80s Object Oriented DBMS

Impedance mismatch between Objects and relational model

```
Struct person {
  name string;
  hobbies string[];
}
```

person

pid	name	hobbies
1	Chuck	[cheese, surfing]
2	Jane	

```
{
  pid: 1
  name: "Chuck"
  hobbies: [
    "cheese",
    "surfing"
  ]
}
```

Hard to query against
Vender lock-in, language lock-in
NoSQL before it was cool
Mostly addressed by ORMs

56

90s More Money

Sybase licensed by Microsoft for x86 → *Microsoft SQLServer*

MySQL created

Postgres adds support for SQL → *PostgreSQL*

SQLite created.
Most popular DB in the world
Second most popular software library in the world (after zlib)

57

00s Data Warehouses aka OLAP

Companies accumulate transaction data, want to analyze
Want high throughput (OLAP)
Distributed shared nothing databases
Mostly closed source
Limited scalability (tens of nodes)

PostgreSQL begats Netezza, Greenplum
MonetDB from CWI
Vertica from MIT
Paraccel →RedShift

58

00s The Internet

Internet companies want scalability over all else
Give up most of RDBMS features for scalability
  Schema, relational model, ACID, SQL, strong consistency
Rise of eventually consistent NoSQL stores
Open Source

BigTable, Dynamo, Hbase, MongoDB, Couchbase, Cassandra

59

10s NewSQL

Eventual consistency is difficult to program against
Scalable distributed DBs that support ACID, SQL
Shared nothing via partitioning (sharding)

Industry: Google Spanner, MemSQL, SAP Hana
Academic →Company: Hstore →VoltDB, Hyper,
Open Source: Yugabyte, Cockroach

60

10s Cloud Databases

Cloud infrastructure = Scalable resource provisioning

Early on containerize open source databases
Provides devops, provisioning
Doesn't best leverage cloud
AWS Postgresql, Amazon RDS

Cloud native databases
Optimized for distributed storage (shared disk)
Decouples storage and execution, scales independently
Lots of connectors to the "data lake"
Presto, Snowflake, Apache Drill, AWS Redshift, Azure Cloud

61

10s Specialized DBs

Graph systems
Graph API over relational (node, edge) data
CIDR 2015 GRAL: fast DBMS > specialized graph engines
Adage: graph traversals are joins. DBs are *very good* at joins
Neo4J, Dgraph, Graphbase, Giraph, GraphX

Time series
Timestamp + a few attributes
Custom indexing, execution for high performance
Timescale, InfluxDB, clickhouse,

62

10s Specialized DBs

Streaming
Events arrive in a "stream"
Querying over an infinitely growing table

Evolution of streaming systems
Continuous queries: *SASE, Telegraph, STREAM*
Sliding windows: *Aurora, Oracle CQL*
Scalability, Best-effort semantics: *Twitter Storm*
Out of order, state: *Flink, Spark Streaming, Kafka, Materialized*

63

10s DBs for X

Classic DB architecture are still good bones
RDBMS is mostly commodity
Market shift toward "massive scale" and "for X"

Where X is
Ads
Video
Astronomy
ML
Vis
…

Rule of thumb
10x better not enough
>50x better

64

At the end of the day

Main Market: DB ***Apps & Services***

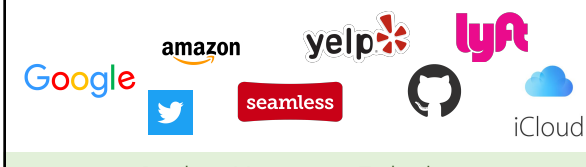
Salesforce: CRM
IBM, SAP, Oracle: CRM, supply chain, inventory, HR
Adobe: marketing, advertisement
Google, FB, Yandex, Twitter: the web, search
Microsoft: productivity, cloud
Amazon: commerce, cloud

Each company created major DBMSes

65

At the end of the day

Main Market: DB ***Apps & Services***



Database Management Technology

66

Your Tasks for next class

Submit reviews for Thursday

Submit Assignment 0 by 9/13 11:59PM EST

w6113.github.io

67