Introduction

Introduction to Databases

CompSci 316 Spring 2019

DUKE
COMPUTER SCIENCE

Welcome to

CompSci 316: Introduction to Database Systems!! Spring 2019

I THINK WE SHOULD BUILD AN SQL DATABASE.
UH-OH
DOES HE UNDERSTAND WHAT HE SAID OR IS IT SOMETHING HE SAW IN A TRADE MAGAZINE AD?
WHAT COLOR DO YOU WANT THAT DATABASE?
I THINK MAUVE HAS THE MOST RAM.
S. Adams E-mail: SCOTTADAMS@AOL.COM
© 1995 United Feature Syndicate, Inc.(NYC)

Acknowledgement: Thanks to Prof. Jun Yang for the course material of CompSci 316!

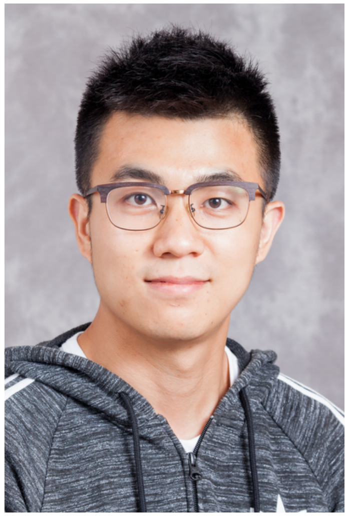
About us: instructor

- Instructor: Sudeepa Roy
  - At Duke CS since Fall 2015
  - A proud member of "Duke Database Devils" group ☺ https://sites.duke.edu/duke_dbgroup/
  - PhD. UPenn, Postdoc: U. of Washington
  - Research interests:
    - "data"
    - data management, database theory, data analysis, causality and explanations, uncertain data, data provenance, crowdsourcing, … .

About us: TAs

Grad TA

UTAs



Zhengjie

PhD student in DB group

(wrote RATest)

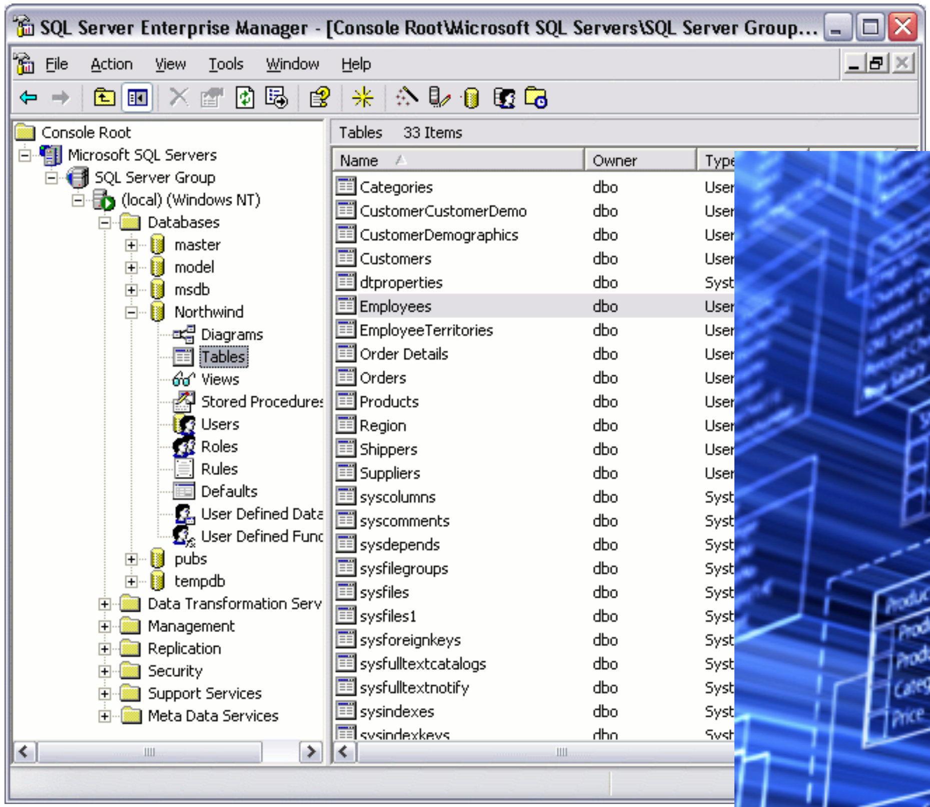
Sarah

Elliott

Both CompSci 316 veterans!

What comes to your mind…

…when you think about “databases”?

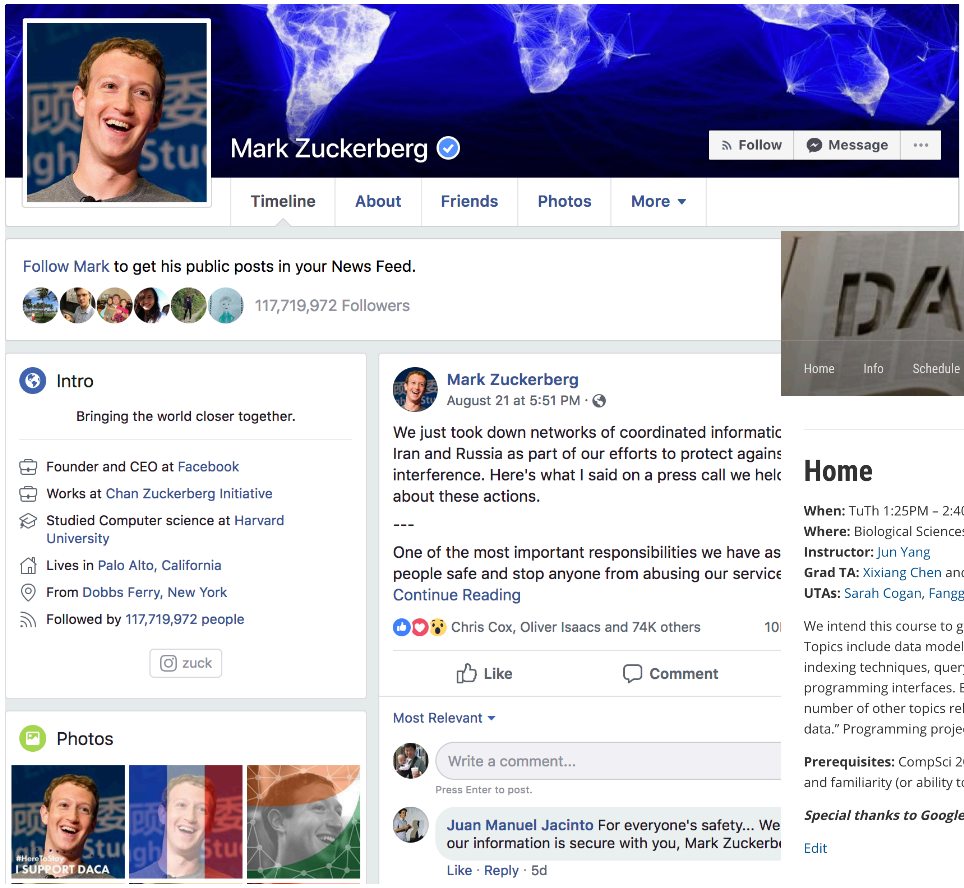


Tables 33 Items

Name	Owner	Type
Categories	dbo	User
CustomerCustomerDemo	dbo	User
CustomerDemographics	dbo	User
Customers	dbo	User
dtproperties	dbo	Syst
Employees	dbo	User
EmployeeTerritories	dbo	User
Order Details	dbo	User
Orders	dbo	User
Products	dbo	User
Region	dbo	User
Shippers	dbo	User
Suppliers	dbo	User
syscolumns	dbo	Syst
syscomments	dbo	Syst
sysdepends	dbo	Syst
sysfilegroups	dbo	Syst
sysfiles	dbo	Syst
sysfiles1	dbo	Syst
sysforeignkeys	dbo	Syst
sysfulltextcatalogs	dbo	Syst
sysfulltextnotify	dbo	Syst
sysindexes	dbo	Syst
sysindexkeys	dbo	Syst

http://www.quackit.com/pix/database/tutorial/dbms_sql_server.gif
http://webstoresltd.com/wp-content/uploads/2013/06/database-design.jpg

But these use databases too…



Follow Mark to get his public posts in your News Feed.

117,719,972 Followers

Bringing the world closer together.

Founder and CEO at Facebook

Works at Chan Zuckerberg Initiative

Studied Computer science at Harvard University

Lives in Palo Alto, California

From Dobbs Ferry, New York

Followed by 117,719,972 people

Mark Zuckerberg
August 21 at 5:51 PM ·

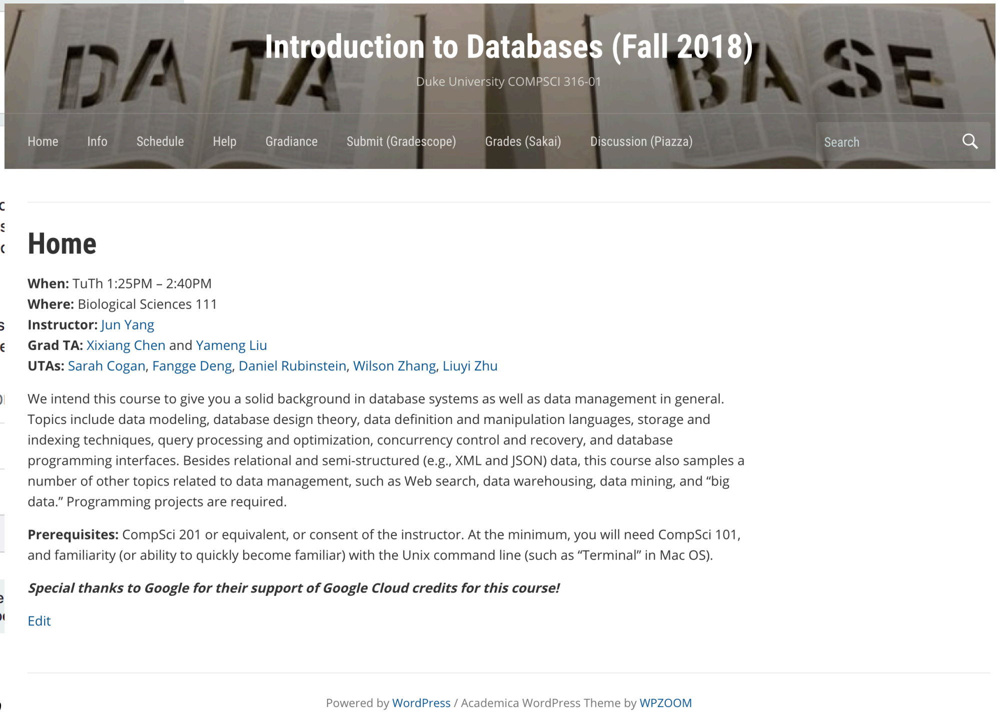
We just took down networks of coordinated informatic Iran and Russia as part of our efforts to protect agains interference. Here's what I said on a press call we helc about these actions.

One of the most important responsibilities we have as people safe and stop anyone from abusing our service
Continue Reading

Chris Cox, Oliver Isaacs and 74K others 10

Juan Manuel Jacinto For everyone's safety... We our information is secure with you, Mark Zuckerbe
Like · Reply · 5d

Facebook uses MySQL to store posts, for example



Home

When: TuTh 1:25PM – 2:40PM
Where: Biological Sciences 111
Instructor: Jun Yang
Grad TA: Xixiang Chen and Yameng Liu
UTAs: Sarah Cogan, Fangge Deng, Daniel Rubinstein, Wilson Zhang, Liuyi Zhu

We intend this course to give you a solid background in database systems as well as data management in general. Topics include data modeling, database design theory, data definition and manipulation languages, storage and indexing techniques, query processing and optimization, concurrency control and recovery, and database programming interfaces. Besides relational and semi-structured (e.g., XML and JSON) data, this course also samples a number of other topics related to data management, such as Web search, data warehousing, data mining, and "big data." Programming projects are required.

Prerequisites: CompSci 201 or equivalent, or consent of the instructor. At the minimum, you will need CompSci 101, and familiarity (or ability to quickly become familiar) with the Unix command line (such as "Terminal" in Mac OS).

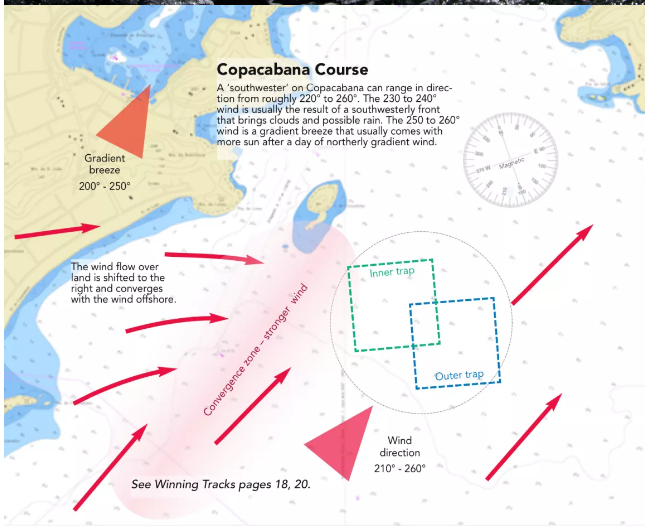
Special thanks to Google for their support of Google Cloud credits for this course!

Edit

Powered by WordPress / Academica WordPress Theme by WPZOOM

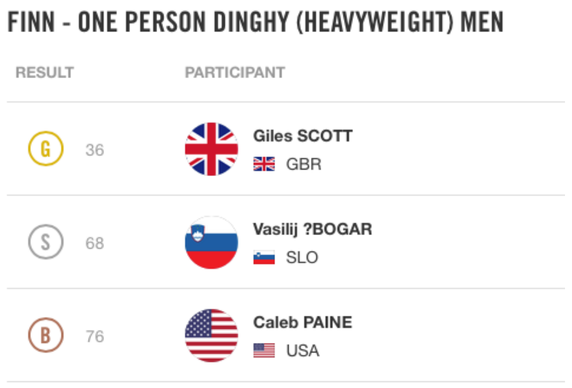
WordPress uses MySQL to manage components of a website (pages, links, menus, etc.)

Data → Gold (ok, Bronze)


Copacabana Course
A 'southwester' on Copacabana can range in direction from roughly 220° to 260°. The 230 to 240° wind is usually the result of a southwesterly front that brings clouds and possible rain. The 250 to 260° wind is a gradient breeze that usually comes with more sun after a day of northerly gradient wind.
Gradient breeze 200° - 250°
The wind flow over land is shifted to the right and converges with the wind offshore.
Convergence zone - stronger wind
Inner trap
Outer trap
Wind direction 210° - 260°
See Winning Tracks pages 18, 20.

… The three years of gathering and analyzing data culminated in what U.S. Sailing calls their "Rio Weather Playbook," a body of critical information about each of the seven courses only available to the U.S. team…

— FiveThirtyEight, **"Will Data Help U.S. Sailing Get Back On The Olympic Podium?"**
Aug 15, 2016

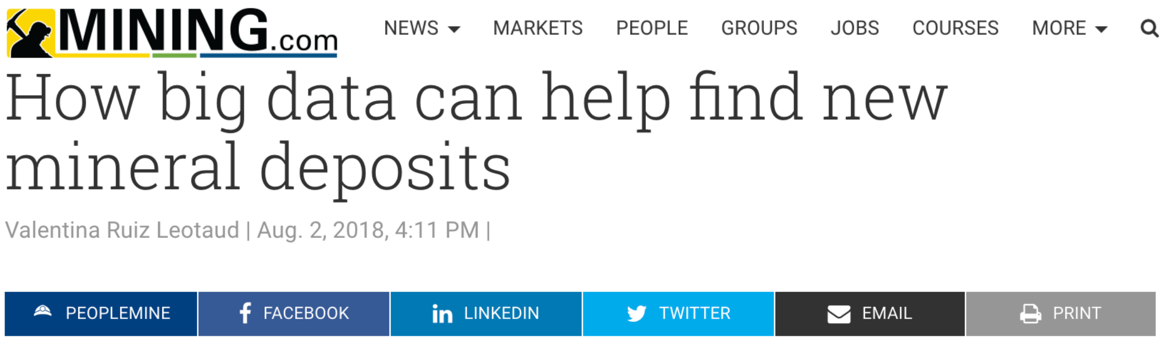


FINN - ONE PERSON DINGHY (HEAVYWEIGHT) MEN

RESULT		PARTICIPANT
G	36	Giles SCOTT GBR
S	68	Vasilij ?BOGAR SLO
B	76	Caleb PAINE USA

http://fivethirtyeight.com/features/will-data-help-u-s-sailing-get-back-on-the-olympic-podium/

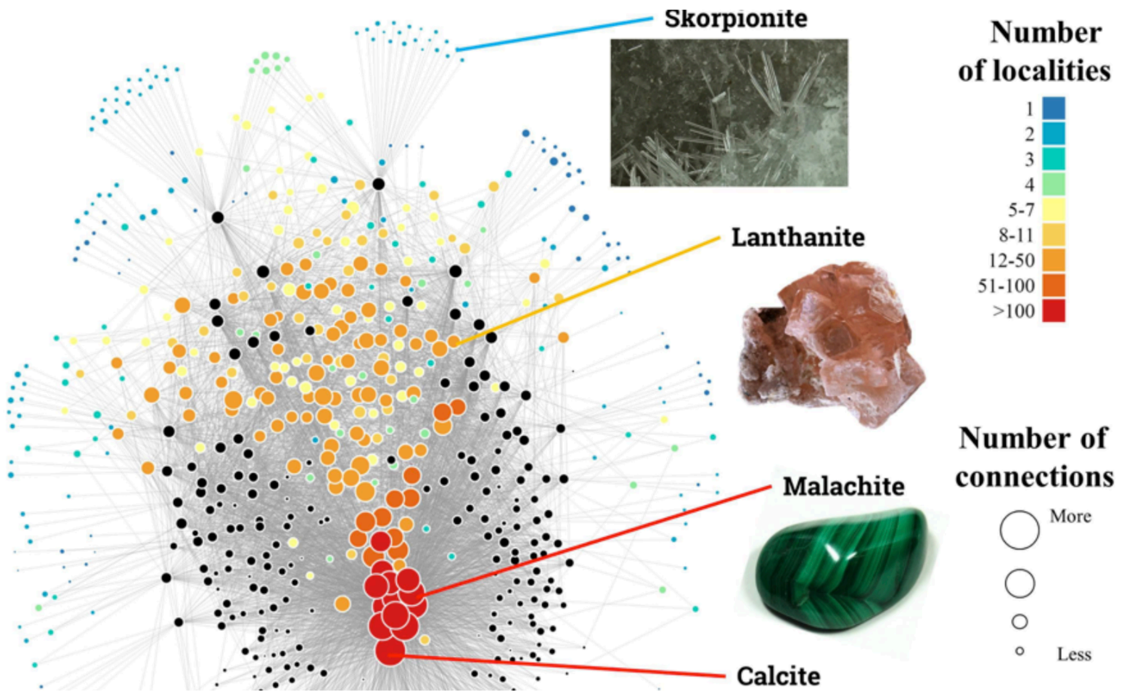
Data → Resources


MINING.com NEWS MARKETS PEOPLE GROUPS JOBS COURSES MORE

How big data can help find new mineral deposits

Valentina Ruiz Leotaud | Aug. 2, 2018, 4:11 PM |

PEOPLEMINE FACEBOOK LINKEDIN TWITTER EMAIL PRINT



Scientists from the Deep Carbon Observatory in the U.S. published a study where they report the first application to mineralogy of network theory, commonly used in the analysis of the spread of disease, terrorist cell connections, or Facebook connections.

The study appeared in American Mineralogist *and it shows how the **application of big data analysis to mineralogy can help predict minerals missing from those known to science, as well as where to find new deposits.***

http://www.mining.com/big-data-can-help-find-new-mineral-deposits/

Data → fun and profit

The New York Times

When Sports Betting Is Legal, the Value of Game Data Soars

A trader working at William Hill, an international sports betting book, in Las Vegas.
Bridget Bennett for The New York Times

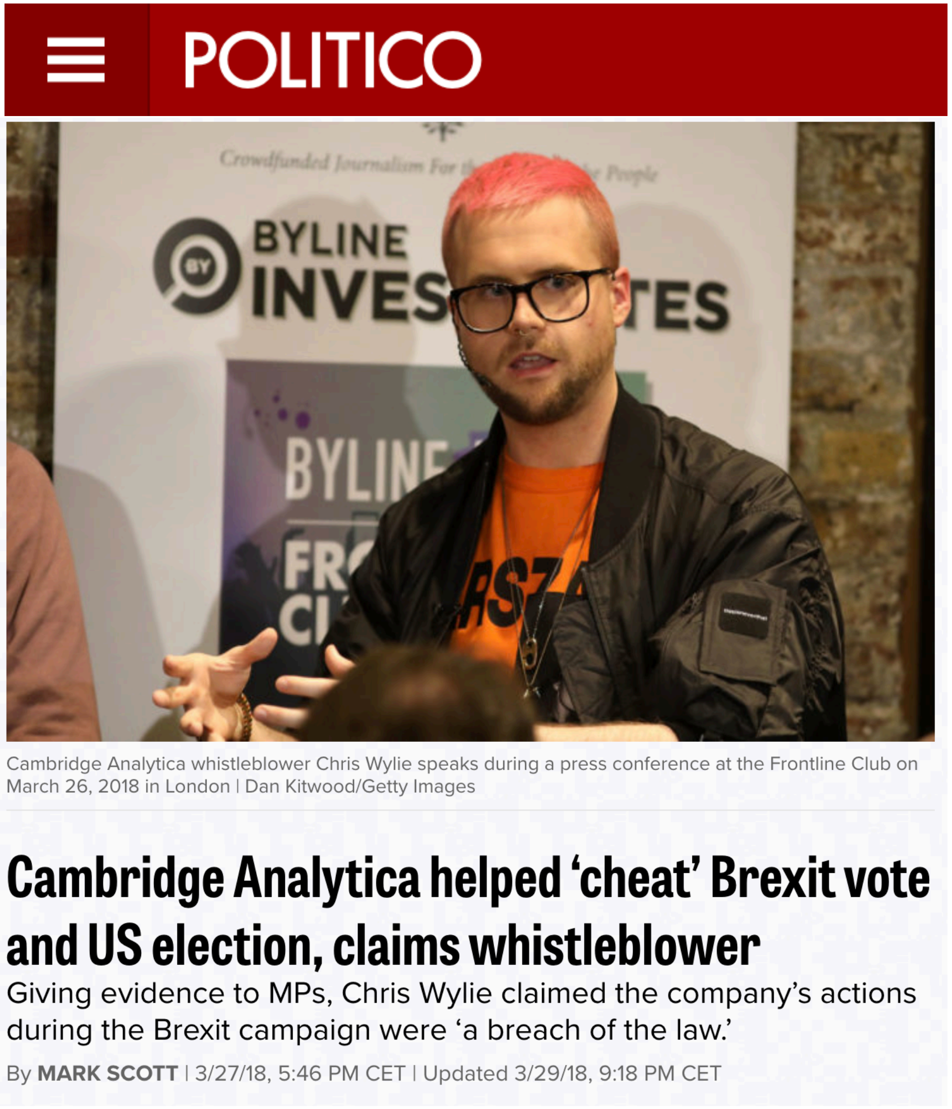
https://www.nytimes.com/2018/07/02/sports/sports-betting.html

Every weekend during soccer season in Britain, security personnel find them in stadiums, tapping furiously at their phones or talking nonstop into a mic — mysterious customers often wearing hoodies to conceal earpieces and their identity…

The unofficial data scouts — or data thieves, depending on who is describing them — are quickly ejected once they are discovered.

The fleeting data they are collecting — the minutia of what is happening in the game — is the lifeblood of sports betting, *perhaps the most crucial and valuable element of the entire industry.*

Data → power


POLITICO

Cambridge Analytica whistleblower Chris Wylie speaks during a press conference at the Frontline Club on March 26, 2018 in London | Dan Kitwood/Getty Images

Cambridge Analytica helped 'cheat' Brexit vote and US election, claims whistleblower

Giving evidence to MPs, Chris Wylie claimed the company's actions during the Brexit campaign were 'a breach of the law.'

By **MARK SCOTT** | 3/27/18, 5:46 PM CET | Updated 3/29/18, 9:18 PM CET

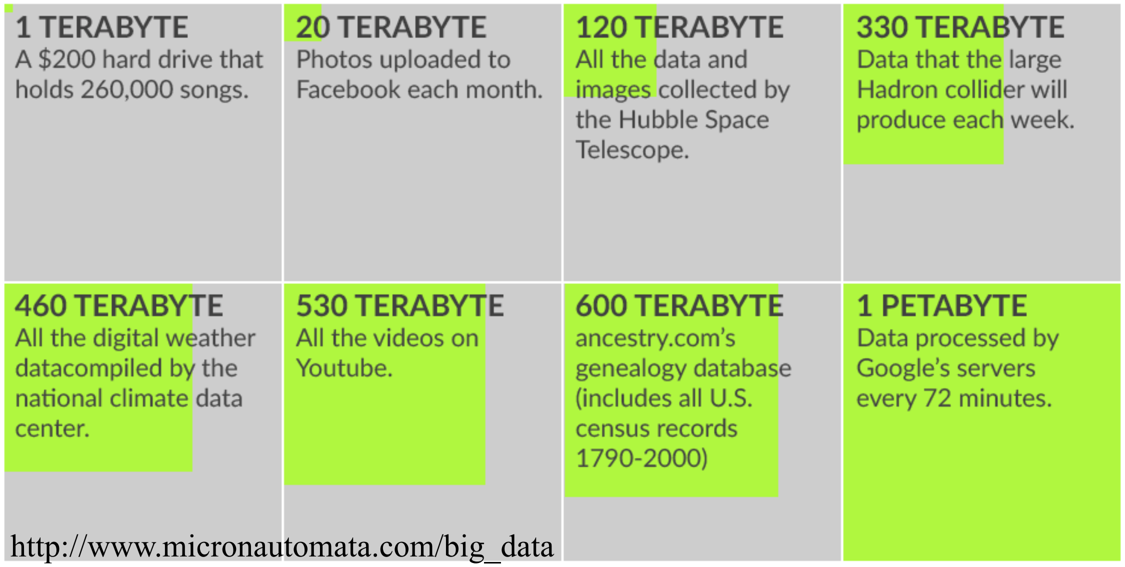
Chris Wylie, the former director of research at Cambridge Analytica, which has been accused of illegally collecting ***online data of up to 50 million Facebook users.***

"He added that a Canadian business with ties to Cambridge Analytica's parent company, SCL Group, also provided analysis for the Vote Leave campaign ahead of the 2016 Brexit referendum. This research, Wylie said, likely breached the U.K.'s strict campaign financing laws and may have helped to sway the final Brexit outcome.

https://www.politico.eu/article/cambridge-analytica-chris-wylie-brexit-trump-britain-data-protection-privacy-facebook/

Challenges

- Moore's Law:
  Processing power doubles every 18 months
- But amount of data doubles every 9 months
  - Disk sales (# of bits) doubles every 9 months
  - Parkinson's Law:
    Data expands to fill the space available for storage



1 TERABYTE A $200 hard drive that holds 260,000 songs.	**20 TERABYTE** Photos uploaded to Facebook each month.	**120 TERABYTE** All the data and images collected by the Hubble Space Telescope.	**330 TERABYTE** Data that the large Hadron collider will produce each week.
460 TERABYTE All the digital weather datacompiled by the national climate data center.	**530 TERABYTE** All the videos on Youtube.	**600 TERABYTE** ancestry.com's genealogy database (includes all U.S. census records 1790-2000)	**1 PETABYTE** Data processed by Google's servers every 72 minutes.

http://www.micronautomata.com/big_data

Moore's Law reversed

Time to process all data
doubles every 18 months!

- Does your attention span double every 18 months?
    - No, so we need smarter data management and processing techniques

Democratizing data (and analysis)

- Democratization of data: more data—relevant to you and the society—are being collected
  - "Smart planet": sensors for phones and cars, roads and bridges, buildings and forests, …
  - "Government in the sunshine": spending reports, school performance, crime reports, corporate filings, campaign contributions, …
- But few people know how to analyze them
- You will learn how to help bridge this divide

Misc. course info

- Website: https://sites.duke.edu/compsci316s2019/
- Course info; tentative schedule and reference sections in the book; lecture slides, assignments, help docs, …
- Book: *Database Systems: The Complete Book*, by H. Garcia-Molina, J. D. Ullman, and J. Widom. 2nd Ed.
- Programming: VM required; $50 worth of credits for VMs in the cloud, courtesy of Google
- Q&A on Piazza
- Grades, sample solutions on Sakai
- Watch your email for announcements
- Office hours to be posted

Grading

A mix of absolute grading and curves

Guarantees:

[90%, 100%]	A- / A / A+
[80%, 90%)	B- / B / B+
[70%, 80%)	C- / C / C+
[60%, 70%)	D
[0%, 60%)	F

Class topper gets A+

At least 30% in the A range

At least the next 30% in the B range

- Scale will not go upwards—mistake would be mine alone if I made an exam too easy

Duke Community Standard

- See course website for link
- Group discussion for assignments is okay (and encouraged), but
    - Acknowledge any help you receive from others
    - Make sure you “own” your solution
- All suspected cases of violation will be aggressively pursued

Course load

- Four homework assignments (35%)
  - Gradiance: immediately and automatically graded
  - Plus written and programming problems; submit through Gradescope
- Course project (25%)
  - Details to be given in the third week of class
- Midterm and final (20% each)
  - Open book, open notes
  - No communication/Internet whatsoever
  - Final is comprehensive, but emphasizes the second half of the course

Projects from past years

- RA: next-generation relational algebra interpreter
  - You may get to try it out for Homework #1!
- *Duke Conversations*: a website to help manage the program, which hosts informal dinners with faculty and students to foster engagement on campus
- *PantryPals*: a social network for amateur cooks, as an Android app
- *LegiToken*: a website to help users research on ICOs (Initial Coin Offerings) by consolidating information from multiple sources and social media

Projects from past years

- *Partners for Success Tutoring App*: connecting volunteer tutors to Durham teachers and students
- Congress Talking Points: analyses (sentiment, similarity, etc.) of speeches by members of the Congress
- wikiblocks (vimeo.com/147680387): find visualizations for Wiki pages
- FarmShots: help farmers with analysis of satellite images
- FoodPointsMaster: tracks balance & spending habit

More past examples

- Pickup Coordinator: app for coordinating carpool/pickups
- FriendsTracker app: where are my friends?
- Duke Schedulator: ditch ACES—plan visually!
- SensorDB: manage/analyze sensor data from forest
- K-ville tenting management

Your turn to be creative

http://www.yummymummyclub.ca/sites/default/files/styles/large/public/field/image/teaching_kids_creative_skills.jpg

So, what is a database system?

From Oxford Dictionary:

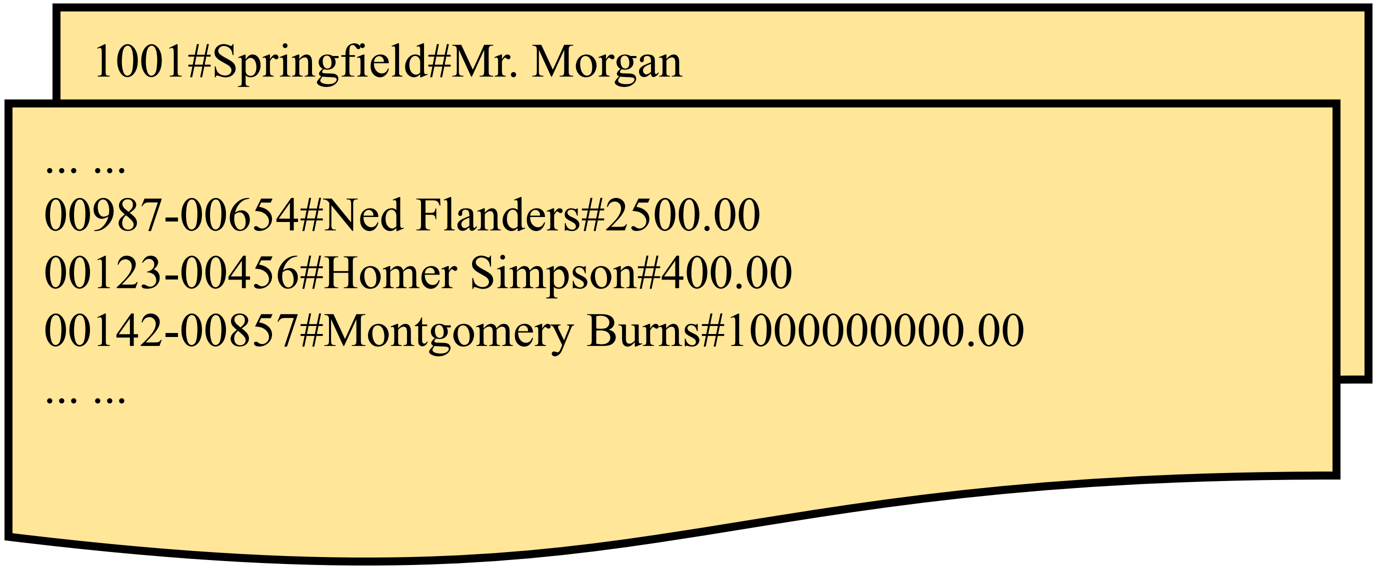
- Database: an organized body of related information
- Database system, DataBase Management System (DBMS): a software system that facilitates the creation and maintenance and use of an electronic database

What do you want from a DBMS?

- Keep data around (persistent)
- Answer questions (queries) about data
- Update data

- Example: a traditional banking application
  - Data: Each account belongs to a branch, has a number, an owner, a balance, …; each branch has a location, a manager, …
  - Persistency: Balance can't disappear after a power outage
  - Query: What's the balance in Homer Simpson's account? What's the difference in average balance between Springfield and Capitol City accounts?
  - Modification: Homer withdraws $100; charge accounts with lower than $500 balance a $5 fee

Sounds simple!

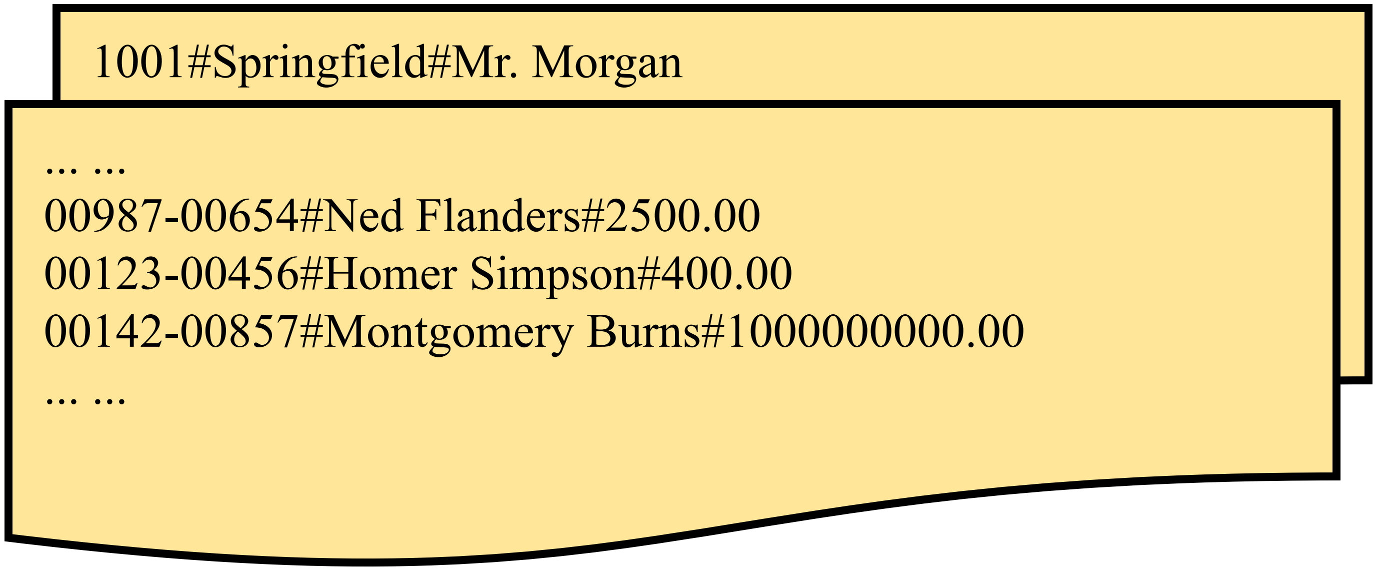


```
1001#Springfield#Mr. Morgan
```

```
… …
00987-00654#Ned Flanders#2500.00
00123-00456#Homer Simpson#400.00
00142-00857#Montgomery Burns#1000000000.00
… …
```

- Text files
- Accounts/branches separated by newlines
- Fields separated by #'s

Query by programming



```
1001#Springfield#Mr. Morgan
```

```
… …
00987-00654#Ned Flanders#2500.00
00123-00456#Homer Simpson#400.00
00142-00857#Montgomery Burns#1000000000.00
… …
```

- What’s the balance in Homer Simpson’s account?
- A simple script
  - Scan through the accounts file
  - Look for the line containing “Homer Simpson”
  - Print out the balance

Query processing tricks

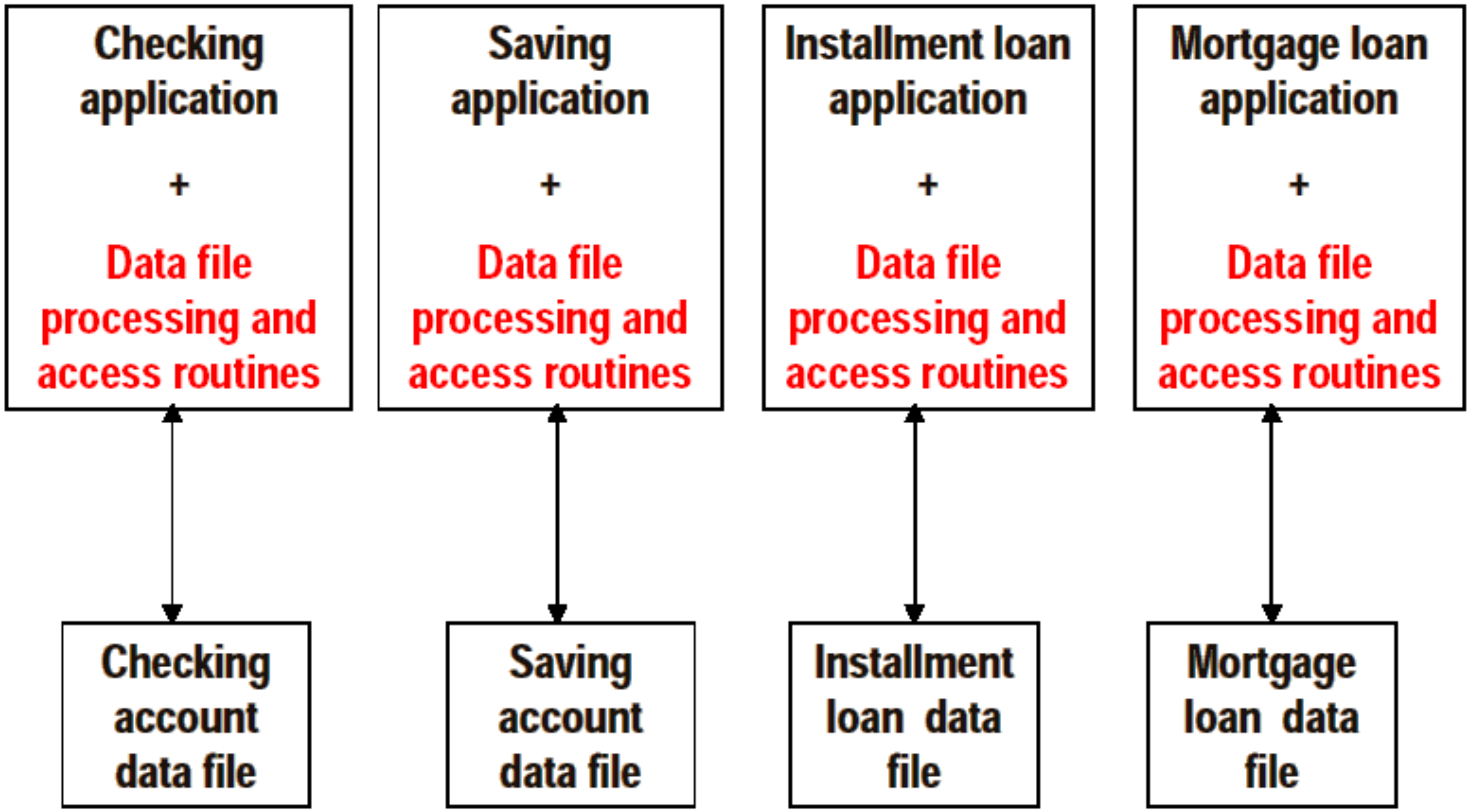
- Tens of thousands of accounts are not Homer’s
  - ☞Cluster accounts by owner’s initial: those owned by “A...” go into file A; those owned by “B...” go into file B; etc. → decide which file to search using the initial
  - ☞Keep accounts sorted by owner name → binary search?
  - ☞Hash accounts using owner name → compute file offset directly
  - ☞Index accounts by owner name: index entries have the form ⟨*owner_name, file_offset*⟩ → search index to get file offset
  - ☞And the list goes on…

What happens when the query changes to: *What’s the balance in account 00142-00857*?

Observations

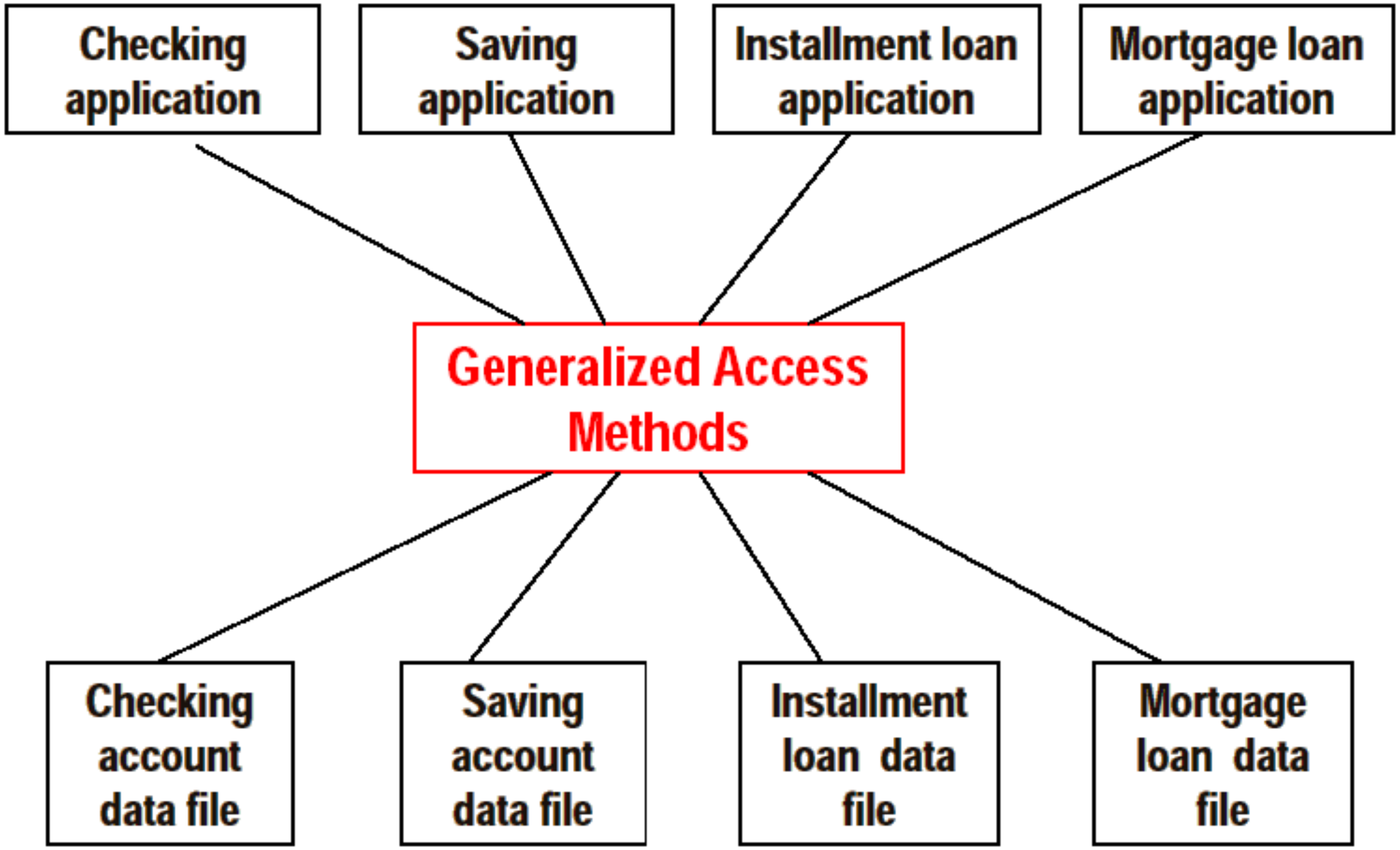
- There are many techniques—not only in storage and query processing, but also in concurrency control, recovery, etc.
- These techniques get used over and over again in different applications
- Different techniques may work better in different usage scenarios

The birth of DBMS – 1


Checking application
+
Data file processing and access routines
Saving application
+
Data file processing and access routines
Installment loan application
+
Data file processing and access routines
Mortgage loan application
+
Data file processing and access routines
Checking account data file
Saving account data file
Installment loan data file
Mortgage loan data file

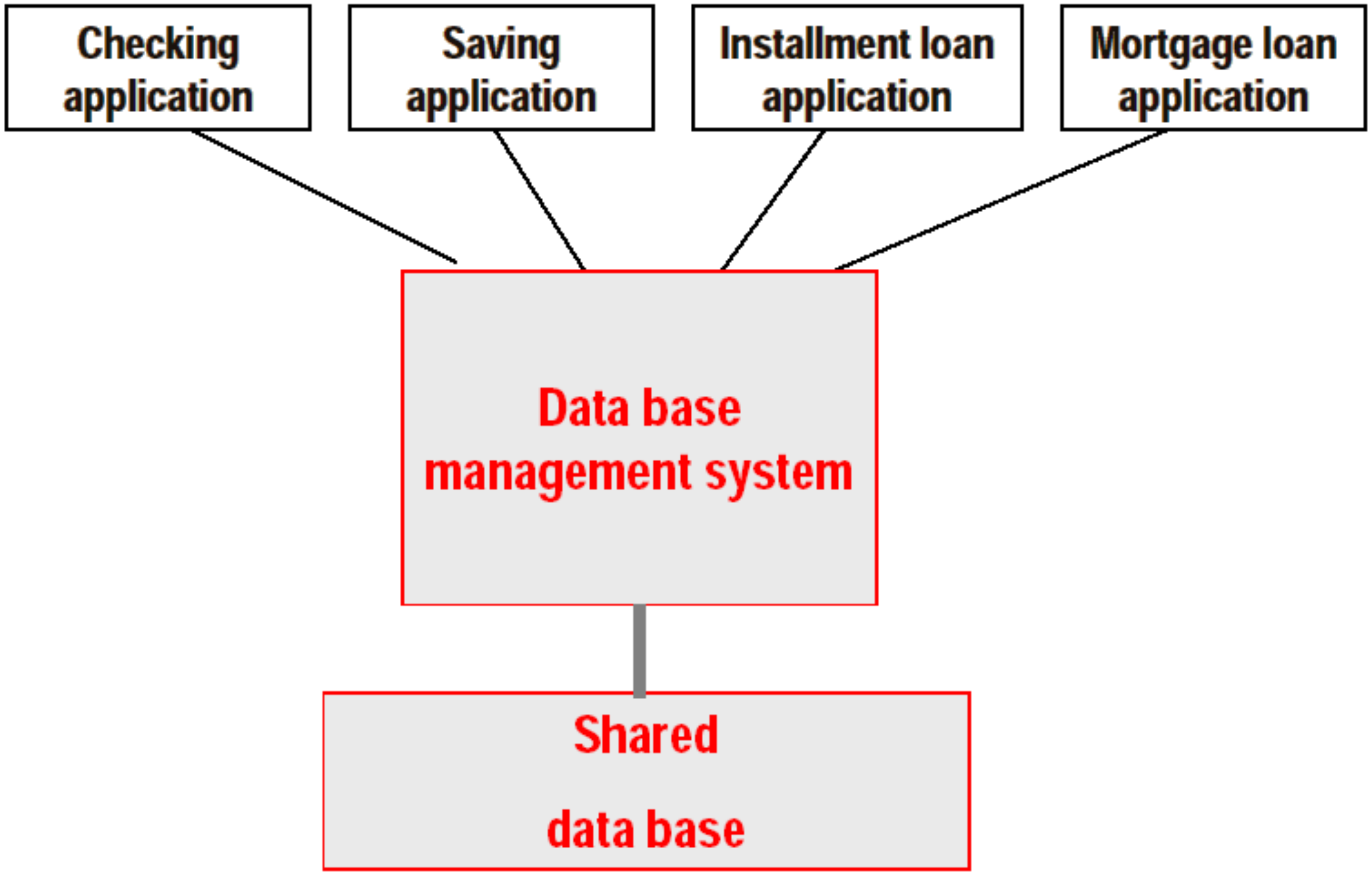
From Hans-J. Schek's *VLDB* 2000 slides

The birth of DBMS – 2


Checking application
Saving application
Installment loan application
Mortgage loan application
Generalized Access Methods
Checking account data file
Saving account data file
Installment loan data file
Mortgage loan data file

From Hans-J. Schek's *VLDB* 2000 slides

The birth of DBMS – 3


Checking application
Saving application
Installment loan application
Mortgage loan application
Data base management system
Shared data base

From Hans-J. Schek's *VLDB* 2000 slides

Early efforts

- "Factoring out" data management functionalities from applications and standardizing these functionalities is an important first step
  - CODASYL standard (circa 1960's)
  - ☞ Bachman got a Turing award for this in 1973

- But getting the abstraction right (the API between applications and the DBMS) is still tricky

CODASYL

- Query: Who have accounts with 0 balance managed by a branch in Springfield?
- Pseudo-code of a CODASYL application:

```
Use index on account(balance) to get accounts with 0 balance;
For each account record:
  Get the branch id of this account;
  Use index on branch(id) to get the branch record;
  If the branch record’s location field reads “Springfield”:
    Output the owner field of the account record.
```

- Programmer controls “navigation”: accounts → branches
  - How about branches → accounts?

What’s wrong?

- The best navigation strategy & the best way of organizing the data depend on data/workload characteristics

With the CODASYL approach

- To write correct code, programmers need to know how data is organized physically (e.g., which indexes exist)
- To write efficient code, programmers also need to worry about data/workload characteristics

☞Can’t cope with changes in data/workload characteristics

The relational revolution (1970's)

- A simple model: data is stored in relations (tables)
- A declarative query language: SQL

```
SELECT Account.owner
FROM Account, Branch
WHERE Account.balance = 0
AND Branch.location = 'Springfield'
AND Account.branch_id = Branch.branch_id;
```

- Programmer specifies what answers a query should return, but not how the query is executed
- DBMS picks the best execution strategy based on availability of indexes, data/workload characteristics, etc.

☞Provides physical data independence

Physical data independence

- Applications should not need to worry about how data is physically structured and stored
- Applications should work with a logical data model and declarative query language
- Leave the implementation details and optimization to DBMS
- The single most important reason behind the success of DBMS today
  - And a Turing Award for E. F. Codd in 1981

Standard DBMS features

- Persistent storage of data
- Logical data model; declarative queries and updates → physical data independence
    - Relational model is the dominating technology today

☞What else?

DBMS is multi-user

- Example

```
get account balance from database;
if balance > amount of withdrawal then
    balance = balance - amount of withdrawal;
    dispense cash;
    store new balance into database;
```

- Homer at ATM1 withdraws $100
- Marge at ATM2 withdraws $50
- Initial balance = $400, final balance = ?
    - Should be $250 no matter who goes first

Final balance = $300

Homer withdraws $100:

```
read balance; $400



if balance > amount then
    balance = balance - amount; $300
    write balance; $300
```

Marge withdraws $50:

```
read balance; $400
if balance > amount then
    balance = balance - amount; $350
    write balance; $350
```

Final balance = $350

Homer withdraws $100:

```
read balance;                 $400

if balance > amount then
    balance = balance - amount;   $300
    write balance;   $300
```

Marge withdraws $50:

```
read balance;                 $400



if balance > amount then
    balance = balance - amount;   $350
    write balance;                $350
```

End of Lecture 1

Concurrency control in DBMS

- Similar to concurrent programming problems?
  - But data not main-memory variables
- Similar to file system concurrent access?
  - Lock the whole table before access
    - Approach taken by MySQL in the old days
    - Still used by SQLite (as of Version 3)
  - But want to control at much finer granularity
    - Or else one withdrawal would lock up all accounts!

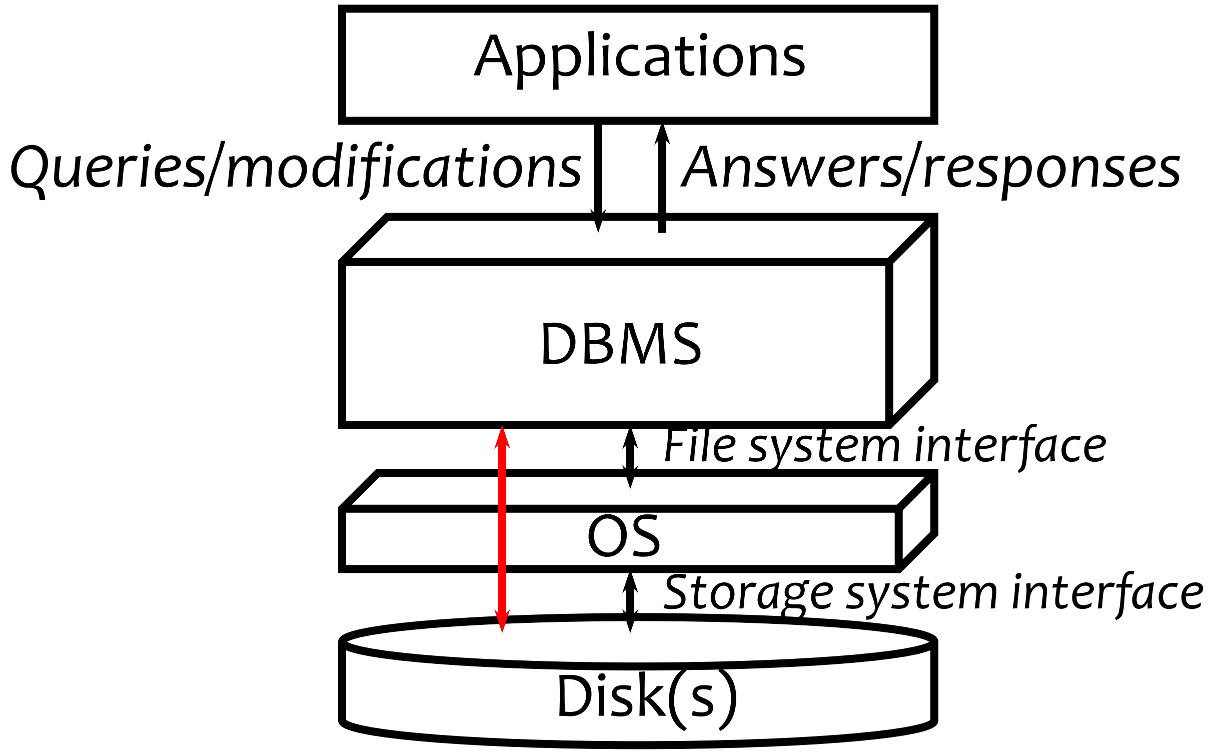
Recovery in DBMS

- Example: balance transfer
  decrement the balance of account X by $100;
  increment the balance of account Y by $100;
- Scenario 1: Power goes out after the first instruction
- Scenario 2: DBMS buffers and updates data in memory (for efficiency); before they are written back to disk, power goes out
- How can DBMS deal with these failures?

Standard DBMS features: summary

- Persistent storage of data
- Logical data model; declarative queries and updates → physical data independence
- Multi-user concurrent access
- Safety from system failures
- Performance, performance, performance
  - Massive amounts of data (terabytes~petabytes)
  - High throughput (thousands~millions transactions/hour)
  - High availability (≥ 99.999% uptime)

Standard DBMS architecture



- Much of the OS may be bypassed for performance and safety
- We will be filling in many details of the DBMS box throughout the semester

AYBABTU?

"Us" = relational databases

- Most data are not in them!
  - Personal data, web, scientific data, system data, …

- Text and semi-structured data management
  - XML, JSON, …
- "NoSQL" and "NewSQL" movement
  - MongoDB, Cassandra, BigTable, HBase, Spanner, HANA…
- This course will look beyond relational databases

Use of AYBABTU inspired by Garcia-Molina
Image: http://upload.wikimedia.org/wikipedia/en/0/03/Aybabtu.png

Course components

- Relational databases
    - Relational algebra, database design, SQL, app programming
- Semi-structured data
    - Data model and query languages, app programming, interplay with relational databases
- Database internals
    - Storage, indexing, query processing and optimization, concurrency control and recovery
- Advanced topics (TBD)
    - Parallel data processing/MapReduce, data warehousing and data mining, Web search and indexing, etc.

Announcements (Jan. 10)

- Email me if you have registration questions
  - But questions of possible general interest should always go to Piazza instead!

- Next class we will do relational algebra—the first of many query languages we shall learn this semester

- More info on course Gradiance, VM setup, and Google credits will be announced on piazza soon